

LIST L

;

RAM ALLOCATION AND EQUATES

0000 = .0000 ORG 0

;

DATA THAT DOESN'T REQUIRE INITIALIZATION.

;

0000 = .0020	DS	32	4. RESERVE 32 BYTES SCRATCH PAD AREA
.0020 = .0040	PIHPIR	DS	64 : PATH/DIRECTION POINTER TABLE
.0040 = .0001	JSDATA	DS	1 : JOYSTICK DATA
.0061 = .0001	SCOREC	DS	1 : SCORE COUNT (USED TO INCREMENT SCORE BY SCRSPD)
.0062 = .0008	SNDPTR	DS	8 : POINTER TO SOUND DATA TABLE
.006A = .0001	BDEADC	DS	1 : BUCK DEAD COUNTER
.006B = .0002	ADDR	DS	2 : 2 BYTE POINTER TO SHAPE
.006D = .0002	ADDRM	DS	2 : 2 BYTE POINTER TO MASK
.006E = .0001	BIT	DS	1 : X AXIS BIT POSITION WITHIN
.0070 = .0001	BYTE	DS	1 : THE BYTE
.0071 = .0001	HIGHT	DS	1 : JUST WHAT IT SAYS
.0072 = .0002	MASK	DS	2 : POINTER FOR MASK USE
.0074 = .0002	SCREEN	DS	2 : POINTER FOR SCREEN USE
.0076 = .0002	SHAPE	DS	2 : POINTER FOR SHAPE USE
.0078 = .0001	TIMER	DS	1 : 1 BYTE TIMER
.0079 = .0001	YCOOD	DS	1
.007A = .0001	KBFLG	DS	1
.007B = .0001	LVLFLG	DS	1
.007C = .0001	MEANEG	DS	1
.007D = .0001	SPEEDC	DS	1
.007E = .0001	WIDITH	DS	1
.007F = .0001	WIDTH	DS	1

;

DATA THAT NEEDS TO BE INITIALIZED TO 0 BEFORE EACH PLAY

;

.0080 = .0001	LASTBS	DS	1 : LAST BUCK USED
.0081 = .0001	BSECNT	DS	1 : COUNTER FOR BUCK FIRE REPEAT
.0082 = .0001	POLCNT	DS	1 : POLE REPEAT COUNTER
.0083 = .0001	SAUCNT	DS	1 : SAUCER DELAY COUNT (TIME REMAINING)
.0084 = .0001	HOPCNT	DS	1 : HOPPER DELAY COUNT (TIME REMAINING)
.0085 = .0004	SNDAGE	DS	4 : SOUND AGE
.0089 = .0004	SDSTAT	DS	4 : SOUND STATUS
.008D = .0001	POLEND	DS	1 : POLE FOUND FLAG/POLE X
.008E = .0001	SSCORE	DS	1 : USED FOR CALCULATING ADDITIONAL SHIPS
.008F = .0001	SCRATCH	DS	1 : USED FOR FUEL FLASH
.0090 = .0001	HFRCNT	DS	1 : HOPPER FIRE COUNTS
.0091 = .0001	PERCNI	DS	1 : POLE FIRE COUNTS
.0092 = .0001	MERCNI	DS	1 : MOTHER ZORBA FIRE COUNT
.0093 = .0001	STRCNT	DS	1 : STAR COUNT

;

DATA TO BE INITIALIZED BEFORE EACH PLAY

;

.0094 = .0001	SCRSET	DS	1 : SCROLL SET COUNTER
.0095 = .0001	SCRCNT	DS	1 : SCROLL COUNT
.0096 = .0001	MTNSPD	DS	1 : MOUNTAIN SPEED
.0097 = .0001	MISIRI	DS	1 : START OF MOUNTAIN DATA CARD #'S
.0098 = .0001	MTNCNT	DS	1 : CURRENT COUNT OF MOUNTAIN
.0099 = .0001	SCRSPD	DS	1 : SCROLL SPEED
.009A = .0001	BUCKX	DS	1 : BUCK SHIP X POSITION
.009B = .0001	BUCKY	DS	1 : BUCK SHIP Y POSITION
.009C = .0003	FULAMT	DS	3 : AMOUNT OF FUEL REMAINING

listing circa 1983  
 Donated by  
 Charlie Keler  
 5-15-18

1 ; DATA TO BE INITIALIZED PRIOR TO EVERY NEW LEVEL OF PLAY

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	
4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57		
5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57			
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57				
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57					
8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57						
9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57							
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57								
11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57									
12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57										
13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57											
14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57												
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57													
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57														
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57															
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																	
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																		
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																			
22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																				
23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																					
24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																						
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57																							
26	27	28	2																																																				

1  
2  
3 \* ATARI HARDWARE EQUATES (111)

4  
5  
6 = E80A RANDOM EQU \$E80A  
7 = D40A WSYNC EQU \$D40A  
8 = D401 CHACTL EQU \$D401  
9 = D402 CHABASE EQU \$D402  
10 = D403 DLISHL EQU \$D403  
11 = D404 DLISHT EQU \$D404  
12 = D405 DMACTL EQU \$D405  
13 = E808 AUDCTL EQU \$E808  
14 = C01F CONSL EQU \$C01F  
15 = E800 AUDEL F01 \$E800  
16 = E801 AUDC1 EQU \$E801  
17 = E802 AUDC2 EQU \$E802  
18 = E803 AUDC3 EQU \$E803  
19 = E804 AUDC4 EQU \$E804  
20 = E805 AUDC5 EQU \$E805  
21 = E806 AUDC6 EQU \$E806  
22 = E807 AUDC7 EQU \$E807  
23 = C01D GRACTL EQU \$C01D  
24 = D40E NMEN EQU \$D40E  
25 = D302 PACTL EQU \$D302  
26 = D303 PBCTL EQU \$D303  
27 = D40Z PMBASE EQU \$D40Z  
28 = 0206 VDSLST EQU \$0206  
29 = 0202 VVBLKI EQU \$0202  
30 = 0800 RTNADD EQU \$800 : RETURN ADDRESS FOR GOSUB IN PATH ROUTINE  
31 = 1000 RELSCR EQU \$1000 : RELOCATED SCROLL COLOR ADDR  
32 = E80F SKCTL EQU \$E80F  
33 = D40B VCOUNT EQU \$D40B  
34 = E80E SKSTAT EQU \$E80E  
35 = E809 KBCODE EQU \$E809  
36

37 \* RANDOM OTHER EQUATES

38 = 0600 DBJTBL EQU \$600

39 0918 = 2FFD ORG \$2FFD  
40 2FFD FF DB \$FF  
41 2FFF 0040 DW START

42 \* SYSTEM STARTUP

43 = A000 = 6000 ORG \$4000

44 = 6000 START

45 \* CLEAR HIGH SCORE AND SCORE

46 6000 AD0709 LDA COLDFG  
47 6003 D9A5 CMP #\$A5  
48 6005 D00A ^6011 BNE COLDST  
49 6007 AD0809 LDA COLDFG+1

```

1 600A C95A      CMP     #$5A
2 600C D008 ^6011  BNE     COL0ST
3 600E 4CE568      JME     EINIT
4 6011 A200      LDA     #0
5 6013 A203      LDX     #3
6 6015      CLERHI
7 6015 9D0909      STA     HISCOR,X
8 6018 9D0D09      STA     GSORE,X
9 601B EA      DEX
10 601C 10FZ ^6015   BPL     CLERHI
11 601E A9A5      LDA     #$A5
12 6020 8D0709      STA     COLDFG
13 6023 A95A      LDA     #$5A
14 6025 8D0809      STA     COLDFG+1
15 6028 40E548      JMP     EINIT ; NO PWR UP INIT
16 602B      DRIVER
17 602B 203474      JSR     JOYSTK ; READ JOYSTICK
18 602E 204A60      JSR     CONVSP ; MAKE MTNSPD EFFECT SCROLL SPEED
19 6031 203D6B      JSR     MTNWRIK ; MOVE MOUNTAINS
20 6034 206163      JSR     SOUND ; SOUNDS
21 6037 209C4D      JSR     ESFIRE ; LET FIRE FASTER
22 603A 207E6E      JSR     MOVE ; MOVE OBJECTS ON DISPLAY
23 603D 206163      JSR     SOUND ; SOUNDS AGAIN
24 6040 20236C      JSR     BSWORK ; BS PROCESSING
25 6043 20E168      JSR     POLES ; DO POLE PROCESSING
26 6046 20CD6B      JSR     SAUCER ; DO SAUCER PROCESSING
27 6049 207E62      JSR     HOPPER ; DO HOPPER PROCESSING
28 604C 202664      JSR     COLISN ; DO COLLISION DETECTION
29 604F 20D863      JSR     SCORE ; DISPLAY SCORE
30 ;           JSR     UF0DSP ; ***??**?
31 6052 201962      JSR     FUEL ; DO FUEL PROCESSING
32 6055 209A61      JSR     HOPFIR ; HOPPER FIRE ROUTINE
33 6058 20F9A1      JSR     POLEFIR ; POLE FIRE ROUTINE
34 605B 20FC60      JSR     MOTHER ; PROCESS MOTHER ZORBA
35 605E 20B960      JSR     STARS ; PROCESS STARS
36 11 6061 200000      JSR     PAUSE ; CHECK FOR PROGRAM PAUSE
37 6064 209B63      JSR     BOUND
38 6067 4C2B60      JMP     DRIVER
39
40 606A 48      CONVSP     PHA
41 606B 8A      TXA
42 606C 48      PHA
43 606D A599      LDA     SCRSPD ; KLUDGE FOR ATARI CONVERSION
44 606F A208      LDX     #8
45 6071 11E8340     CVLOOP     CMP     SCONVE-1,X
46 6074 9007 ^607D     BCC     CONVOK
47 6076 F005 ^607D     BEQ     CONVOK
48 6078 CA      DEX
49 6079 D0F61 C6071     BNE     CVLOOP
50 607B A201      LDX     #$.1
51 607D 8ECA00     CONVOK     STX     SPEED
52 6080 68      PLA
53 6081 AA      TAX
54 6082 68      PLA
55 6083 60      RTS
56 1138837848 SCONVE     DB     $A0, $98, $88, $78, $68, $58, $48, $38

```

ROUTINE DISPLAYS UFO'S REMAINING TO BLAST

```
1 608C UFODSP
2 608C A5A6 LDA UEOCNT
3 608E F020 ^60B0 BEQ UEOXT
4 6090 C913 CMP #19
5 6092 901D ^60B1 BCC UEPY
6 6094 A956 LDA #22 OR 64
7 6096 8D4420 STA $2064
8 6099 A912 LDA #18
9 609B 8500 UEOOK STA 0
10 609D A900 LDA #0
11 609F A213 LDX #19
12 60A1 9D6420 UEOIX STA $2064,X
13 60A4 CA DEX
14 60A5 F009 ^60B0 BEQ UEOXT
15 60A7 E400 CPX 0
16 60A9 10E6 ^60A1 BNE UEOIX
17 60AB A915 LDA #21
18 60AD ACA160 JMP UEOIX
19 60B0 60 RTS
20 60B1 A000 LDY #0
21 60B3 8C6420 STY $2064
22 60B6 4C9B60 JMP UEOOK
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

1 ROUTINE CREATES STARS AT CONSTANT TIME PERIODS, BUT  
 2 DEPEN ENOUGH THAT THE DISPLAY CONSTANTLY HAS  
 3 LOTS OF STARS IN IT.

```

5 60B9 STARS
6 60B9 A5AC LDA SYSTAT : SEE IF SPACE SCENE
7 60BB 2902 AND #2
8 60BD D001 ^60C0 BNE STR20 ; YES - CREATE STARS
9 60BE STR10
10 60BE 60 RTS
11 60C0 STR20
12 60C0 E693 INC STRCNT
13 60C2 A593 LDA STRCNT
14 60C4 4A LSR A
15 60C5 90E8 ^60BE ROR STR10 ; NO
16 60C7 A209 LDX #9 ; TRY TO CREATE A STAR
17 60C9 20E06C JSR CREATE
18 60CC D0F1 ^60BE BNE STR10 ; NO ROOM - GO
19 60CE STR30
20 60CE A00A8 LDA RANDOM ; X=64-191 (128 POSITINS)
21 60D1 30FB ^60CE BMI STR30
22 60D3 18 CLC
23 60D4 6240 ADC #64
24 60D6 990106 STA ORJTBL+1,Y
25 60D8 08 PHP ; SAVE SIGN OF X
26 60DA A00A8 LDA RANDOM ; GET PATH BASE 1-5
27 60DD 2907 AND #$7
28 60DE E0F9 ^60DA BEQ STR40
29 60E1 C904 CMP #6 ; ONLY 5
30 60E3 B0P5 ^60DA BCS STR40
31 60E5 19 CLC ; NORMALIZE
32 60E6 A913 ADC #19 ; ACTUAL PATH #19-20-30
33 60E8 28 PLP ; GET SIGN OF X
34 60E9 3003 ^60FF BMI STR50 ; NEGATIVE X IS GET 20-24
35 60F1 18 CLC
36 60F2 6206 ADC #6 ; POSITIVE X IS GET 25-29
37 60F3 200EAD STR50 JSR SETPTH ; SET PATH
38 60F4 A934 LDA #54 ; Y CENTERED (ALMOST)
39 60F5 990206 SIA ORJTBL+2,Y
40 60F6 A930 LDA #30 ; SPEED
41 60F8 990506 STA ORJTBL+5,Y
42 60FB 60 RTS
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

1 ROUTINE BASICALLY JUST MAKES SURE THAT WHEN MOTHER  
2 ZORBA IS SUPPOSED TO BE OUT THERE, SHE IS.

```

4 60FC MOTHER
5 60FC A5AC LDA SYSTAT ; SEE IF SUPPOSED TO BE A ZORBA
6 60FF 2904 AND #4
7 6100 D001 ^6108 BNE MOTH20 ; YES
8 6102 MOTH10 RTS
9 6102 60
10 6103 MOTH20
11 6103 A000 LDY #0 ; SEE IF MOTHER ZORBA OUT THERE
12 6105 MOTH30
13 6105 B90006 LDA OBJTBL,Y
14 6108 C906 GMR #6
15 610A F040 ^614C BEQ MOTH70 ; GOT ONE - SEE IF TO FIRE
16 610C 98 TYA ; TRY NEXT
17 610D 18 CLC
18 610E 6908 ADC #8
19 6110 A8 TAY
20 6111 C000 CRY #0
21 6113 B0F0 ^6105 BNE MOTH30
22
23 ; ZORBA'S SUPPOSED TO BE OUT THERE AND SHE AINT.. TRY TO PUT HER THERE.
24
25 6115 A206 LDX #6
26 6117 20E06C JSR CREATE
27 611A D0E6 ^6102 BNE MOTH10 ; CAN'T - OH WELL
28 611C A5AC LDA SYSTAT ; SEE IE. SPACE/GROUND
29 611E 2902 AND #2
30 6120 F01F ^6141 BEQ MOTH60 ; GROUND PATHS DIFFERENT THAN SPACE PATHS
31 6122 ADOAES LDA RANDOM ; SELECT PATH 15/16
32 6125 2005 ^6120 BMI MOTH40 ; USE PATH 15
33 6127 A910 LDA #16 ; USE PATH 16
34 6129 4C2E61 JMP MOTH50
35 612C A90F MOTH40
36 612E A920 LDA #15
37 612E 200E6D JSR SETPATH
38 6131 A980 LDA #$80 ; X, Y, SPEED
39 6133 990106 STA OBJTBL+1,Y
40 6136 A920 LDA #144
41 6138 990206 STA OBJTBL+2,Y
42 613B A5A5 LDA MTRSPD
43 613D 990506 STA OBJTBL+5,Y
44 6140 60 RTS
45
46 6141 MOTH60
47 6141 ADOAES LDA RANDOM ; GROUND PATHS USE SAUCER PATHS FOR NOW
48 6144 2903 AND #3
49 6146 18 CLC
50 6147 6908 ADC #3
51 6149 4C2E61 JMP MOTH50
52
53 ; ZORBA OUT THERE - SEE IF TO FIRE
54
55 614C MOTH70
56 614C B90706 LDA OBJTBL+7,Y
57 614E 2910 AND #$10
58 6151 D001 ^6154 BNE MOTH90 ; ATTACKING - MAYBE
59 6153 MOTH80

```

```

1 6153 40 RTS
2 6154 MOTH80
3 6154 A592 LDA MERCNT
4 6156 E003 ^615B BEQ MOT100
5 6158 C692 DEC MERCNT
6 615A 40 RTS
7 615B MOT100
8 615B 840B STY 11 ; SAVE INDEX TO ZORBA
9 615D A5AB LDA MFRBLY ; RESET COUNTER FOR NEXT TIME
10 615E 8592 STA MERCNT
11 6161 B90306 LDA OBJTBL+3,Y ; IF ZORBA SIZE<2, NO FIRE ALLOWED
12 6164 C902 CMP #2
13 6166 90EB ^6153 BCC MOTH80 ; FORGET IT - RAM BUCK INSTEAD?
14 6168 A204 LDX #4 ; START ENEMY SHOT
15 616A 90E06C JSR CREATE
16 616E B0E4 6153 BNE MOTH80 ; NO ROOM
17 616F B90C LDA #12 ; USE PATH12 (LIKE HOPPER FIRE)
18 6171 200E6D JSR SETETH
19 6174 A60B LDX 11 ; SET UP TO GET ZORBA X+INDEX DATA
20 6176 B00306 LDA OBJTBL+3,X ; ZORBA SIZE USED FOR INDEX
21 6178 990306 STA OBJTBL+3,Y ; SHOT SIZE = ZORBA SIZE
22 617C AA TAX
23 617D B0A466 LDA MTRHTR,X ; X OFFSET TO CENTER
24 6180 A60B LDX 11 ; ADD TO ZORBA X
25 6182 18 CLC
26 6183 7D0106 ADC OBJTBL+1,X
27 6184 990106 STA OBJTBL+1,Y ; SHOT X
28 6189 B00206 LDA OBJTBL+2,X ; Y=ZORBA Y
29 618C 990206 STA OBJTBL+2,Y
30 618F A920 LDA #20 ; SPEED
31 6191 990504 STA OBJTBL+5,Y
32 6194 A903 LDA #3 ; SOUND SAME AS HOPPER FIRE
33 6196 20A963 JSR SNDINI
34 6199 40 RTS
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

1 ROUTINE CHECKS SYSTEM STATUS TO SEE IF HOPPERS ARE  
 2 TO FIRE. IF SO, THE HERDLY IS USED TO DETERMINE HOW  
 3 FREQUENTLY THE WILL FIRE

```

 5 619A    HOPFIR
 6 619A    A5AC      LDA      SYSTAT      ; SEE IF HOPPERS FIRE THIS ROUND
 7 619C    1004    ^61A4      BPL      HOPF10      ; NO
 8 619E    A590      LDA      HRCNT      ; SEE IF TO FIRE THIS TIME
 9 61A0    E003    ^61A5      BEQ      HOPF20      ; YES
10 61A2    C690      DEC      HRCNT
11 61A4    HOPF10
12 61A4    60      RTS
13 61A5    HOPF20
14 61A5    A010      LBY      #16      ; TRY TO FIND A HOPPER
15 61A7    HOPF30
16 61A7    B90006      LDA      OBJTBL,Y
17 61AA    C903      CMP      #3
18 61AC    E00C    ^61BA      BEQ      HOPF40      ; GOT ONE
19 61AE    HOPF35
20 61AE    98      TYA
21 61AE    18      CLC
22 61B0    4908      ADC      #8
23 61B2    A8      TAY
24 61B3    C980      CMP      #\$80
25 61B5    D0F0    ^61A7      BNE      HOPF30
26 61B7    4CF461      JMP      HOPF50      ; NONE - EXIT
27 61BA    HOPF40
28 61BA    B90106      LDA      OBJTBL+1,Y      ; FIRE ONLY IF IN DISPLAY AREA
29 61BD    C928      CMP      #40      ; X MUST BE 40-215
30 61BE    90E9    ^61AE      BCC      HOPF35
31 61C1    B9D7      CMP      #215
32 61C3    B0E9    ^61AE      BCS      HOPF35
33 61C5    941F      STY      \$1F      ; SAVE Y
34 61C7    A204      LDX      #4      ; 4 IS ENEMY SHOT
35 61C9    20E06C      JSR      CREATE      ; TRY TO CREATE A SHOT
36 61CC    D0D6    ^61A4      BNE      HOPF10      ; NO ROOM - FORGET IT
37 61CE    A90C      LDA      #12      ; THESE ARE ON PATH12
38 61D0    200E6D      JSR      SETPTH
39 61D3    A81F      LDX      \$1F      ; GET INDEX TO SOURCE OF SHOT
40 61D5    BD0106      LDA      OBJTBL+1,X      ; GET X+4 FOR START
41 61D8    18      CLC
42 61D9    6904      ADC      #4
43 61DB    990106      STA      OBJTBL+1,Y
44 61DE    BD0206      LDA      OBJTBL+2,X      ; Y
45 61E1    990206      STA      OBJTBL+2,Y
46 61E4    BD0306      LDA      OBJTBL+3,X      ; SIZE=HOPPER SIZE
47 61E7    990306      STA      OBJTBL+3,Y
48 61EA    A980      LDA      #\$80      ; SPEED
49 61EC    990506      STA      OBJTBL+5,Y
50 61EF    A903      LDA      #3      ; ENEMY FIRE SOUND
51 61F1    20A863      JSR      SNDINI
52 61F4    HOPF50
53 61F4    A5A9      LDA      HERDLY
54 61F6    8590      STA      HRCNT
55 61F8    60      RTS

```

1 ; POLE FIRE ROUTINE CHECK TO SEE IF THEY SHOULD FIRE SHOTS

 2 41F2  
 3 41F9 A5AC LDA SYSTAT  
 4 41F8 2960 AND #\$60  
 5 41F0 F004 ^6205 BEQ POLE10 ; NOT TO FIRE  
 6 41FF A591 LDA PFRCNT  
 7 4201 F003 ^6206 BEQ POLF20 ; FIRE THIS TIME!  
 8 4203 E691 DEC PERCNT  
 9 4205  
 10 6205 POLF10  
 11 6205 60 RTS  
 12 6206  
 13 6206 A010 LDY #16 ; LOOK FOR A POLE  
 14 6208 POLF30  
 15 6208 B90006 LDA OBJTBL,Y  
 16 6208 C901 CMP #1  
 17 620D F00E ^6211 BEQ POLF50 ; GOT ONE  
 18 620F POLF40  
 19 620F 99 TYA  
 20 6210 19 LDI  
 21 6211 6908 ADC #8  
 22 6213 A8 TAY  
 23 6214 C080 CPY #\$80  
 24 6216 D0F0 ^6208 BNE POLF30  
 25 6218  
 26 6218 A5AA LDA PFRDLY ; RESET DELAY TO NEXT FIRE  
 27 6218 8591 STA PERCNT  
 28 621C 40 RTS  
 29 621D  
 30 621D 841F STY \$1F  
 31 621E B90306 LDA OBJTBL+3,Y ; ONLY FIRE ON SIZES 0-6  
 32 6222 C907 CMP #7  
 33 6224 D0F2 ^620F BNE POLF40 ; NO - FORGET THIS ONE  
 34 6226 A00A8 LDA RANDOM ; SEE IF TO FIRE IN/OUT  
 35 6229 4A ISR A  
 36 622A 9042 ^626E BCC POL110 ; FIRE IN  
 37 622C A5AC LDA SYSTAT ; SEE IF OK TO FIRE OUT  
 38 622E 3920 AND #\$20  
 39 6230 F004 ^620F BEQ POLF40 ; NO - TRY ANOTHER ONE  
 40 6232 B90706 LDA OBJTBL+7,Y ; SEE IF THIS A LEFT OR RIGHT POLE  
 41 6235 2908 AND #8  
 42 6237 D005 ^623E BNE POLF70 ; LEFT POLE  
 43 6239  
 44 6239 A20E LDA #14 ; CREATE RIGHT SHOT  
 45 623B 404062 JMP POLF30  
 46 623E  
 47 623E A20D LDA #13 ; CREATE LEFT SHOT  
 48 6240  
 49 6240 351E STA \$1E ; SAVE PATH  
 50 6242 A208 LDX #8 ; TRY TO CREATE POLE SHOT  
 51 6244 20E04C ISR CREATE  
 52 6247 D0CF ^6218 BNE POLF45 ; NO ROOM - FORGET THIS RIGHT NOW  
 53 6249 A51E LDA \$1F ; SET PATH  
 54 624B 200F4D ISR SETPTH  
 55 624E A61F LDX \$1F ; INDEX INTO SOURCE LOCATION  
 56 6250 B00106 LDA OBJTBL+1,X ; SET X  
 57 6253 990106 STA OBJTBL+1,Y  
 58 6256  
 59 6256 BD0206 LDA OBJTBL+2,X ; SET Y=BASE ADDRESS+16  
 60 6256

## POLE FIRE ROUTINE

JF5200 .665

6259 18 CLC  
625A 6910 ADC #16  
625C 990206 STA OBJTBL+2,Y  
625F A980 LDA #\$80 ; SPEED  
6261 990506 STA OBJTBL+5,Y  
6264 A904 LDA #4 ; RAY SOUND  
6266 20A863 JSR SNDINT  
6269 80000000 POLE90 LDI \$1F ; RESTORE Y  
6269 A41F LDY POLE40 ; NEXT  
626B 4C0E62 JMP  
626E 80000000 POL110  
626E A5AC LDA SYSTAT ; SEE IF OK TO FIRE IN  
6270 2240 AND #\$40  
6272 E09B ^620F BEQ POLE40 ; NO - TRY ANOTHER  
6274 B90706 LDA OBJTBL+7,Y ; SEE IF LEFT/RIGHT POLE  
6277 2908 AND #8  
6279 D0BE ^6239 BNE POLE60 ; LEFT - FIRE RIGHT  
627B 4C3E62 JMP POLE70 ; RIGHT - FIRE LEFT

1 ROUTINE TO PROCESS (START) HOPPERS AT PERIODIC INTERVALS ON VARYING PATHS

```

2
3
4 627E HOPPER
5 627E A5AC LDA SYSTAT ; SEE IF THEY EXISTS
6 6280 2910 AND #$10
7 6282 F006 ^628A BEQ HOPP10 ; NO
8 6284 A584 LDA HOPCNT ; SEE IF OK TO START ONE
9 6286 E003 ^628B BEQ HOPP20 ; YEP
10 6288 C684 DEC HOPCNT
11 628A 60 RTS
12 628B HOPP10
13 628B A203 LDX #3 ; TRY TO START A HOPPER
14 628D 20E8AC JSR CREATE
15 6290 D0E8 ^628A BNE HOPP10 ; NO ROOM
16 6292 HOPP30
17 6292 A00AES LDA RANDOM ; GET ONE OF TEN PATHS
18 6295 2901 AND #1 ; PATH 10-11
19 6297 18 CLC
20 6298 690A ADC #10 ; NORMALIZE FOR PATHS (C=0)
21 629A 200E6D JSR SETPTH ; SET PATH IN TABLE
22 629D HOPP40
23 629D A908 LDA #8 ; SET SIZE
24 629F 990306 STA QBJTBL+3,Y
25 62A2 A00AES LDA RANDOM ; SET RANDOM X
26 62A5 990106 STA QBJTBL+1,Y
27 62A8 HOPP50
28 62A8 A5A4 LDA HOPSPD ; SPEED
29 62AA 990506 STA QBJTBL+5,Y
30 62AD A5A3 LDA HOPDLY ; RESET COUNTER
31 62AF 8584 STA HOPCNT
32 62B1 60 RTS ; DONE
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

1 ; ROUTINE DISPLAYS NUMBER OF SHIPS REMAINING, UPDATES ACTUAL  
2 ; EVERY TIME SINCE THIS IS SO FAST (CHARACTER GRAPHICS MODE).  
3

4 62B2 SHIPS  
5 62B2 A5AD LDA SHPLET  
6 62B4 38 SEC  
7 62B5 E901 SBO #1  
8 62B7 F00F, ^62C8 BEQ JAMIT  
9 62B9 C906 CMP #5  
10 62B8 9002 ^62BF BCC SHOK  
11 62B0 A905 LDA #5  
12 62BF AA SHOK TAX  
13 62C0 A914 LDA #20  
14 62C2 9D3520 STA \$2035,X  
15 62C5 CA DEX  
16 62C6 D0EA ^62C2 BNE SHORD  
17 62C8 60 JAMIT RTS  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

```

1      ; ROUTINE DECREMENTS FUEL AND UPDATES DISPLAY
2      ; GRAPH WHENEVER IT CHANGES.
3      ; WAS TIME GRAPH BUT GRAPH NOW REPRESENTS FUEL WHICH DECREMENTS
4      ; AT A RATE INVERSELY TO THE SCROLL SPEED.
5
6      62C9    FUEL
7
8      ; NOW DO FUEL GONE CHECK AND FUEL DECREMENT
9
10     62C9    FUEL03
11     62C9    A59E    LDA    FULAMT+2 ; SEE IF ALREADY OUT OF FUEL
12     62CB    059D    DRA    FULAMT+1
13     62CD    059C    DRA    FULAMT
14     62CF    0001    BNE    FUEL07 ; NOPE
15     62D1    FUEL05
16     62D1    60      RTS
17     62D2    FUEL07
18     62D2    A599    LDA    SCRSPD ; DECREMENT FUEL BY FACTOR OF -SCRSPD
19     62D4    49FF    EOR    #$FF
20     62D6    18      CLC
21     62D7    6901    STA    #1
22     62D9    A214    LDX    #20
23     62DB    20C273   ISR    MULT
24     62DE    8400    STX    0
25     62E0    8501    STA    FULAMT
26     62E2    A59C    LDA    FULAMT
27     62E4    38      SEC
28     62E5    E500    SBC    0
29     62E7    859C    STA    FULAMT
30     62E9    A59D    LDA    FULAMT+1
31     62EB    E501    SBD    1
32     62ED    859D    STA    FULAMT+1
33     62EF    B0F0    RES    FUEL05 ; NO CHANGE IN MSB - JUST EXIT
34     62F1    A59E    LDA    FULAMT+2
35     62F3    E900    SBC    #0
36     62F5    859E    STA    FULAMT+2
37     62F7    B008    6301    BCS    FUEL10
38     62F9    A900    LDA    #0 ; FORCE FUEL=0
39     62FB    859C    STA    FULAMT
40     62FD    859D    STA    FULAMT+1
41     62FF    859E    STA    FULAMT+2
42     6301    FUEL10
43
44      ; --> DISPLAY FUEL GRAPH
45
46      ; FULAMT DETERMINES HOW MUCH FUEL PLAYER HAS. USE MSB_(FUEL+2)
47      ; FOR THE GRAPH WHICH HAS A RANGE OF 00-$77
48
49     6301    A59E    LDA    FULAMT+2
50     6303    4A      ISR    A
51     6304    4A      ISR    A
52     6305    4A      ISR    A
53     6306    C90D    CMP    #13
54     6308    2002    ^630C    BCC    FUISOK
55     630A    A90C    LDA    #12
56     630C    AA      FUISOK    TAX
57     630F    A8      TAY
58
59     630E    F008    ^6318    BEQ    RFHG
60     6310    A9E0    LDA    #32 OR $C0

```

## FUEL GRAPH UPDATE ROUTINE

```

6312 9D4220  FUIP STA $2028+20+6,X
6315 CA DEX
6316 B0FA 6312 BNE FUIP
6318 C8 REHG TNY
6319 28 TYA
631A AA TAX
631B A59E LDA FUIAMT+2
631D 2907 AND #%00000111
631F F007 ^6328 BEQ FUIXX
6321 18 CLC
6322 69D8 ADC #24 DR $C0
6324 9D4220 STA $2028+20+6,X
6327 F8 700 TXN
6329 500D FUIXX CPX #13
632A R009 ^6335 BCS FUXD
632C A9D8 LDA #24 DR $C0
632E 9D4220 STA $2028+20+6,X
6331 F8 TXN
6332 4C2863 JMP FUIXX
20) :--> SET COLOR OF GRAPH
21)
22) : USE FUIAMT+2 AGAIN. $00-$18=RED $19-$30=YELLOW
23) : $31-$77=GREEN (>31)
24)
25) 6335 A59E PUXD LDA FUIAMT+2
26) 6337 C915 CMP #$15
27) 6339 B017 ^6352 BCS TYEL
28) 633B A906 LDA #6
29) 633D 20A863 JSR SNDINI
30) 6340 A58E LDA SCRATCH
31) 6342 4901 FOR #$1
32) 6344 858F STA SCRATCH
33) 6346 D005 ^634D BNE BLNKIT
34) 6348 A90F LDA #$0F
35) 634A 4C5D63 JMP BYEBYE
36) 634D A934 BLNKIT LDA #$34
37) 634F 4C5D63 JMP BYEBYE
38) 6352 C930 TYEL CMP #$30
39) 6354 R005 ^635B BCS TGRN
40) 6356 A91E LDA #$1F
41) 6358 4C5D63 JMP BYEBYE
42) 635B A9E6 TGRN LDA #$E6
43) 635D 8D1409 BYEBYE STA COLOR3
44) 6360 60 RTS
45)
46)
47)
48)
49)
50)
51)
52)
53)
54)
55)
56)
57)

```

## SOUND DRIVER

ROUTINE LOOKS TO SEE IF SOUND GENERATOR IN USE. IF IT IS,  
1. STEPS THROUGH THE FREQUENCIES.

```

4 6361 SOUND
5 6361 A206 LDIX #6 ; INC DATA POINTER INDEX
6 6363 A003 LDY #3 ; INC COUNT/RATE TABLES INDEX
7 6365 SOUN10
8 6365 B98900 LDA SDSTAT,Y ; SEE IF THIS CHANNEL ACTIVE
9 6368 E02B ^6395 BEQ SOUN50 ; NO
10 636A SOUN20
11 636A 208273 JSR INSDPT ; GET NEXT DATA BYTE
12 636D A162 LDA
13 636F F01E ^638F BEQ SOUN40 ; KILL IT
14 6371 C901 CMP #1 ; SEE IF CONTROL BYTE
15 6373 E00E ^6384 BEQ SOUN30 ; YEP
16 6375 9D00E8 STA AUDC1,X ; SET FREQUENCY
17 6378 F98500 LDA SNDAGE,Y ; BIRTHDAY TIME
18 637B 18 CLC
19 637C 6901 ADC #1
20 637E 998500 STA SNDAGE,Y
21 6381 4195A3 JMP SOUN50
22 6384 SOUN30
23 6384 208273 JSR INSDPT ; GET CONTROL BYTE
24 6387 A162 LDA [SNDPTR,X1]
25 6389 9D01E8 STA AUDC1,X ; NOW SET FREQ
26 638C 406A63 JMP SOUN20
27 638F SOUN40
28 638F 998900 STA SDSTAT,Y ; KILL THIS. AND FREE CHANNEL
29 6392 9D01E8 STA AUDC1,X
30 6395 SOUN50
31 6395 CA DEX ; DO ALL
32 6396 CA DEX
33 6397 B8 TIEY
34 6398 10CR ^6365 BPL SOUN10
35 639A 60 RTS

```

## SOUND INITIALIZATION ROUTINE

## SOUND INITIALIZATION ROUTINE

```

1 63E5 4A LSR A
2 63E6 4A LSR A
3 63E7 E501 STA 1
4 63E8 A599 LDA SCRSPD
5 63E9 38 SEC ; SAVE IT
6 63EA E937 SBC #$37 ; GET SCRSPD
7 63EB 18 CLC ; NORMALIZE FOR MIN VALUE
8 63EF 6561 ADC SCOREC
9 63F1 8561 STA SCOREC
10 63F3 6A ROR A ; UPDATE COUNTS
11 63F4 4A LSR A ; KEEP CARRY AND /64
12 63E5 4A LSR A
13 63F6 4A LSR A
14 63F7 4A LSR A
15 63F8 4A LSR A ; SUBTRACT OUT OLD COUNT/64
16 63F9 38 SEC
17 63FA E501 SBC 1 ; X,A IS INCREMENT VALUE FOR SCORE
18 63FC A200 LDX #0
19 63FF 208973 JSR ADDSCR ; ADD TO SCORE
20 ; MAKE CHECK FOR NEW SHIP AWARD
21
22 6401 SCOR01
23 6401 A00E09 LDA CSCORE+1 ; GET CHANGE IN DIGIT 2
24 6404 F8 SED
25 6405 38 SEC
26 6406 E58E SBC SSCORE
27 6408 08 CLD
28 6409 C902 CMP #2
29 640B 9015 ^6422 BCC SCOR05 ; NOT ENOUGH
30 640D F6AD INC SHPLFT ; GIVE ANOTHER SIP
31 640F D002 6413 BNE SCOR02
32 6411 F6AD DEC SHPLFT ; MAX AT FF
33 6413 SCOR02
34 6413 A00E09 LDA CSCORE+1 ; SET SSCORE FOR NEXT PASS
35 6416 29FF AND #$FF
36 6418 858E STA SSCORE
37 641A A90A LDA #$A
38 641C 20A869 JSR SNDINI
39 641F 20B262 JSR SHIPS ; DISPLAY CHANGE
40 6422 SCOR05
41 6422 203F71 JSR SCRUPD ; DISPLAY SCORE
42
43 6425 60 RTS
44 SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 19
45 J:5200 A65
46
47 ; ROUTINE SCANS OBJECT DESCRIPTORS FOR BUCK SHOTS. IF ONE
48 ; IS FOUND, A CHECK IS MADE FOR COLLISIONS WITH ANY DROIDS,
49 ; SAUCERS, OR MOTHER ZORBA HERSELF.
50
51 ; ENTER:
52 ; EXIT:
53 ; REGISTER USAGE:
54 ; $OB = OBJECT LENGTH
55 ; $OC = OBJECT WIDTH
56 ; $OTL = DESTROYED
57

```

```

; $10 = LENGTH
; $11 = WIDTH
; $12 = BUCK SHOT X
; $13 = BUCK SHOT Y
; $1E = OBJECT INDEX SAVE
; $1F = BUCK SHOT INDEX SAVE

```

```

1 6426 COL1SN
2 6426 A000 LDY #0 ; SEE IF ANY BUCK SHOTS
3 6428 COL110
4 6428 B90006 LDA OBJTBL,Y
5 6428 C905 CMP #5
6 642D F00D ^6430 BEQ COL130 ; GOT ONE
7 642F COL120
8 642E 28 TYA ; GO TO NEXT ONE
9 6430 18 CLC
10 6431 6908 ADC #8
11 6433 A8 TAY
12 6434 C900 CMP #$00
13 6436 D0F0 ^6428 BNE COL110
14 6438 4C3065 JMP COL120 ; GO CHECK IF BUCK DEAD DUCK
15 643B COL125
16 643B 60 RTS ; DONE
17
18
19
20 ; GOT BUCK SHOT - SET UP PARAMETERS
21
22 643C COL130
23 643C 841F STY $1E ; SAVE CURRENT INDEX
24 643F B90106 LDA OBJTBL+1,Y ; GET X
25 6441 8512 STA $12
26 6443 B90206 LDA OBJTBL+2,Y ; GET Y
27 6444 38 SEC ; SUBTRACT OUT Y POSITION FUDGE AMOUNT
28 6447 F5B0 SBC FIREDG
29 6449 8513 STA $13
30 644B A20A LDX #10 ; X=TYPE*2 FOR INDEX
31 644D 20506E JSR RCSCL ; GET LENGTH/WIDTH
32 6450 A50B LDA $B ; LENGTH+LENGTH FUDGE TO 10
33 6452 18 CLC
34 6453 65B0 ADC FIREDG
35 6455 8510 STA $10
36 6457 A50C LDA $C ; WIDTH TO 11
37 6459 8511 STA $11
38 ; SEARCH FOR SANGERS/DROIDS(HOPPERS)/MOTHER ZORBA
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

645B A000 LDY #0 ; INIT INDEX
645D B90004 LDA OBJTBL,Y ; SEE IF DROID/SAUCER
6460 C902 CMP #2
6462 E01E ^6482 BEQ COL170 ; SAUCER
6464 C903 CMP #3
6466 E01A ^6482 BEQ COL170 ; DROID
6468 C906 CMP #6
646A E00E ^647A BEQ COL165 ; MOTHER ZORBA
646C COL150
646C 98 TYA ; GO TO NEXT
646D 18 CLC
646E A908 ADC #8
6470 A8 TAY
6471 C000 CPY #$00
6473 BNE COL140
6475 COL160
6475 A41F LDY $1F ; NO MORE - RESTORE FOR NEXT BUCK SHOT
6477 4C2F64 JMP COL120
647A COL165
647A 20AD66 JSR CKCRSH ; SEE IF COLLISION WITH ZORBA
647D FOED ^646C BEQ COL120 ; NO
647F 4C0066 JMP COL120 ; MAYBE
6482 ; GOT SAUCER, DROID, OR ZORBA. GET DISTANCES BETWEEN X,Y COORDINATES AND
6482 ; COMPARE THAT WITH THE LENGTH/WIDTH OF THE OBJECTS
6482 COL170
6482 20AD66 JSR CKCRSH
6485 FOE5 ^646C BEQ COL150 ; NO COLLISION
6487 ; COLLISION DETECTED - CANCEL OBJECT BY FORCING PATH AS EXPLOSION
6487 ; PATH (PATH7). CANCEL BUCK SHOT BY FORCING PATH TO CANCEL PATH
6487 ; (PATH8). IN BOTH CASES, RPTCNT SET TO 0 TO FORCE PATH EXECUTION
6487 ; IMMEDIATELY.
6487 COL110
6487 BE0006 LDX OBJTBL,Y ; SEE WHAT IT WAS FOR SCORE
648A A900 LDA #0
648C 208973 JSR ADDSCR ; ADD TO SCORE
648F A900 LDA #0 ; RESET RPTCNT
6491 990606 STA OBJTBL+6,Y
6494 A907 LDA #7
6496 200E60 JSR SETPTH ; SET OBJECT ON EXPLOSION PATH (PATH7)
6499 A41F LDY #1F ; SET BUCK SHOT INDEX
6499 A900 LDA #0 ; RESET RPTCNT
649D 990606 STA OBJTBL+6,Y
64A0 A908 LDA #8 ; SET ON CANCEL PATH
64A2 200E61 JSR SETPTH
64A5 A5A5 LDA SYSTAT ; DON'T DEC ON ZORBA LEVELS
64A7 2902 AND #%00000010
64A9 E012 ^64BD BEQ DECIT
64AB C6A6 DEC UEOCNT
64AD A5A6 LDA UEOCNT
64AF D010 ^64C1 BNE COL114
64B1 A902 LDA #2
64B3 85A6 STA UEOCNT
64B5 A5AC LDA SYSTAT
64B7 0904 ORA #%00000100
64B9 29F7 AND #%11110111

```

## COLLISION DETECT ROUTINE

```

64BB 85AC STA SYSTAT
64BD 16A6 DECTT DEC UFGENT
64BF E00B ^64CC REG COL115
64C1 COL114
64C1 208C40 JSR UFGDSP ; DISPLAY UFO'S LEFT
64C4 A900 LDA #0 ; START EXPLOSION SOUND
64C6 20A863 JSR SNDINI
64C9 467564 JMP COL160 ; NEXT
; PLAY LEVEL OVER - SET UP PARAMETERS TO START NEXT ONE.
64CC LVI OVR
64CC COL115
64CC >E6AE INC PLYLVL ; BUMP PLAY LEVEL
; DO SOMETHING FANCY TO SHOW PROGRESSION TO NEXT LEVEL
64CE A900 WARP LDA #$0
64D0 A207 LDX #$7
64D2 9100E8 WARPP5 STA AUTO1_X
64D5 CA DEX
64D6 DOEA ^64D2 BNE WARPP5
64D8 A98C LDA #$8C
64DA 8D01E8 STA AUTO1
64DD A98A LDA #$8A
64DF 8D03E8 STA AUTO2
64E2 A0F4 LDY #$E4
64E4 ADOAE8 WARPP1 LDA RANDOM
64E7 8D1102 STA COLOR0
64FA ADOAE8 LDA RANDOM
64FD 8D1209 STA COLOR1
64F0 ADOAE8 LDA RANDOM
64E3 8D1309 STA COLOR2
64F6 ADOAE8 LDA RANDOM
64E9 8D1409 STA COLOR3
64FC ADOAE8 LDA RANDOM
64FF 8D1509 STA COLOR4
6502 A220 LDY #$20 ; WAIT ONE TIEFY
6504 ADOAE8 STA VCOUNT
6507 DOFB ^6504 BNE WAITVB
6509 8C00E8 STY AUTO1
650C 98 TYA
650D 49FF EOR #$FF
650F 8D02E8 STA AUTO2
6512 CA DEX
6513 DOEE ^6504 BNE WAITVB
6515 98 TYA
6516 48 PHA
6517 8A TXA
6518 48 PHA
6519 ADOAE8 CHKVAL LDY RANDOM
651C E005 CPX #5
651E BOF9 ^6519 BCS CHKVAL
6520 209E60 JSR UNSCOL
6523 68 PLA
6524 AA TAX
6525 68 PLA
6526 A8 TAY
6527 88 DEY

```

```

1 6528 88 DEY
2 6529 88 DEY
3 652A 88 DEY
4 652B D0B7 ^64E4 BNE WARP1
5 652D 4C8569 IMP NEWLVL ; GO TO NEW LEVEL
6
7 ; CHECK FOR BUCK SHIP COLLIDED WITH POLES, SAUCERS, ETC.
8 ; LOOKS FOR OBJECTS OF SIZE 0. IF SIZE ONE FOUND, THEN CHECK
9 ; IS MADE TO SEE IF OBJECT DESCRIPTOR LEGITIMATE (PC TYPE > 0).
10 ; IF IT IS, THEN THE OBJECT CAN'T BE A BUCK SHOT. IF ALL
11 ; OF THAT IS OK, THEN A CHECK IS MADE FOR ACTUAL OBJECT COLLISION.
12 6530 COL120
13 6530 A59A LDA BUCKX ; SET UP BUCK SHIP PARAMETERS
14 6532 B512 STA $12
15 6534 A59B LDA BUCKY
16 6536 B513 STA $13
17 6538 A20B LDA #11 ; BUCK LENGTH CONSTANT
18 653A B510 STA $10
19 653C A914 LDA #20 ; BUCK WIDTH CONSTANT
20 653E B511 STA $11
21 6540 A000 LDY #0 ; START FROM HEAD OF OBJTBL
22 6542 COL130
23 6542 B90306 LDA OBJTBL+3,Y ; SEE IF SIZE=0
24 6545 F00A ^6551 BEQ COL150 ; YEP
25 6547 COL140
26 6547 98 TYA ; GO TO NEXT
27 6548 18 CLC
28 6549 6208 ADC #8
29 654B A8 TAY
30 654C C000 CPY #$00
31 654E D0F2 ^6542 BNE COL130
32 6550 COL145 RTS ; NO MORE = DONE
33 6550 40
34 6551 COL150
35 6551 B90006 LDA OBJTBL,Y ; SEE IF OBJECT DESCRIPTOR LEGITIMATE
36 6554 F0E1 ^6547 BEQ COL140 ; NO - FORGET IT
37 6556 C905 CMP #5 ; CAN'T BE BUCK SHOT
38 6558 F0E0 ^6547 BEQ COL140 ; CAN'T BE EXPLOSION
39 655A C907 CMP #7
40 655C F0E9 ^6547 BEQ COL140 ; CAN'T COLLIDE W/ STARS
41 655E C909 CMP #9
42 6560 F0E5 ^6547 BEQ COL140
43 6562 20AD66 JSR CKCRSH ; SEE IF CRASHED
44 6565 F0E0 ^6547 BEQ COL140 ; NOPE
45
46 ; THIS GUY DIED. - START BUCK EXPLOSION
47 ;
48 6567 DEADBK
49 6567 207F6E JSR MOVE ; IN CASE ROOM IS NEEDED
50 656A A207 LDX #7 ; BUCK EXPLOSION IS TYPE 7
51 656C 20E04C ISR CREATE ; ALWAYS WORKS
52 656E D0E6 ^6567 BNE DEADBK ; MOVE THESE GUYS OFF AND MAKE ROOM
53 6571 A59A LDA BUCKX ; SET LOCATION AS BUCK'S
54 6573 990104 STA OBJTBL+1,Y
55 6576 A59B LDA BUCKY
56 6578 990206 STA OBJTBL+2,Y ; SET BUCK EXPLOSION PATH
57 657B A909 LDA #9
58 657D 200E6D JSR SETPTH
59 6580 A900 LDA #0 ; RESET MTNSPD

```

## COLLISION DETECT ROUTINE

```

6582 8596 STA MTNSPD
6584 8599 STA SCRSPD
6586 206A60 JSR CONVSP
6589 A932 LDA #50 ; SET DELAY COUNT
658B 856A STA BDEADC
658D A902 LDA #2 ; BUCK EXPLOSION SOUND
658F 20A863 JSR SNDINI
6592 203272 JSR BSERAS ; ERASE BUCK SHIP/SHADOW
6595 207E6E COL160 JSR MOVE ; JUST CYCLE EXPLOSION/SOUND
6598 206163 JSR SOUND
659B 20DB43 JSR SCORE ; UPDATE SCORE AS REQ
659E A210 LDX #$10 ; DELAY
65A0 A000 LDY #0
65A2 88 COL170 DEY
65A3 D0FD ^65A2 BNE COL170
65A5 CA DEX
65A6 D0EA ^65A2 BNE COL170
65A8 C66A DEC BDEADC
65AA D0E9 ^6595 BNE COL160
65AC C64B DEC SHELET
65AE E014 ^65C4 BEQ COL135 ; GAME OVER
65B0 A52C LDA FULAMT ; SEE IF OUT OF FUEL
65B2 059D ORA FULAMT+1
65B4 059E ORA FULAMT+2
65B6 D009 ^65C1 BNE COL130 ; NO - KEEP GOING.
65B8 A5AE LDA PIYI VI ; JUMP BACK TO FIRST OF 4
65BA 29EC AND #$FC
65BC 85AE STA PIYI VI
65BE 4C8569 IMP NEWI VI
65C1 4C0569 COL180 JMP NEWPLY ; CONTINUE PLAY
65C4 4C0569 COL185
65C4 A940 LDA #Z01000000
65C6 8D0ED4 STA NMEN
65C9 A900 LDA #HIGH_SLIST
65CB 8D0109 STA SDLSTH
65CE A900 LDA #LOW_SLIST
65D0 8D0009 STA SRLSTI
65D3 A9E8 LDA #$F8
65D5 8D0309 STA CHRAS
65D8 A900 LDA #$0
65DA 8578 STA TIMER
65DC 8D1209 STA COLOR1
65DE 8D1309 STA COLOR2
65E2 8D1409 STA COLOR3
65E5 8D1509 STA COLOR4
65E8 8D01E8 STA AUDC1
65EB 8D03E8 STA AUDC2
65EE 8D05E8 STA AUDC3
65F1 8D07E8 STA AUDC4
65F4 A578 BORING LDA TIMER
65F6 8D1109 STA COLOR0
65F9 C9FF CMP #$FF
65FB D0F7 ^65F4 BNE BORING
65FD 4CE568 JMP CINIT

```

; COLLISION DETECTED BETWEEN BUCK SHOT AND MOTHER ZORBA.  
; COLLISION CAN ONLY BE ACCEPTED IF SMALL CENTER REGION

1 ; (REACTOR) DETECTS THE COLLISION. OTHERWISE THE  
2 ; SHOT IS CANCELED, AND AN ENEMY SHOT IS STARTED IN ITS PLACE  
3 ; RETURNING TAKING ONE OF THE STAR PATHS (RANDOM FOR RANDOM  
4 ; RICOCHET). IF COLLISION IS DETERMINED ZORBA BLOWS UP AND  
5 ; THE ROUND IS BROUGHT TO COMPLETION. MTRHTB CONTAINS PARAMETERS  
6 ; FOR FINDING REACTOR GIVEN THE DIFFERENT SIZES OF ZORBAS.

7 6600 COL190  
8 ; LDA OBJTBL+7,Y ; SEE IF ZORBA ATTACKING  
9 ; AND #\$10  
10 ; BNE COL200 ; ALWAYS RETURN REBOUN SHOT HERE  
11 6600 BE0306 ; LDX OBJTBL+3,Y ; GET ZORBA SIZE INDEX  
12 6603 E006 ; CPX #6 ; 6,7,8 ARE NO HITS  
13 6605 B015 ~661C ; BCS COL200  
14 6607 BD0466 ; LDA MTRHTB,X ; GET X OFFSET FOR CENTER OF ZORBA  
15 6608 18 ; CLC  
16 660B 790106 ; ADC OBJTBL+1,Y ; ADD TO X POSITION FOR TEST  
17 660E 38 ; SEC  
18 660F E512 ; SBC \$12 ; GET DISTANCE FROM CENTER  
19 6611 1005 ~6618 ; BPL COL195 ; GET ABS VALUE FOR DISTANCE  
20 6613 49FF ; BDR #\$FF  
21 6615 13 ; CLC  
22 6616 6901 ; ADC #1  
23 6618 COL195  
24 6618 C904 ; CMP #4 ; ACCURACY REQUIREMENT HERE^^  
25 661A 9027 ~6643 ; BCC COL220 ; COLLISION DETECTED - ZORBA'S GONE  
26 ;  
27 ; NO COLLISION DETECTED - REDEFINE SHOT AS ENEMY SHOT ON A REBOUND  
28 ; AND LEAVE ZORBA ALONE  
29 ;  
30 661C COL200  
31 661C A41F ; LDY \$1F ; POINT TO BUCK SHOT TABLE  
32 661E A900 ; LDA #0 ; RESET RRTCNT  
33 6620 990406 ; STA OBJTBL+6,Y  
34 6623 A918 ; LDA #\$18 ; SLOW DOWN SHOT TO MIN  
35 6625 990506 ; STA OBJTBL+5,Y  
36 6628 COL210  
37 6628 ADOAE8 ; LDA RANDOM ; GET RANDOM PATH 31-38  
38 662B 2907 ; AND #\$2  
39 662D 18 ; CLC  
40 662E 691F ; ADC #31  
41 6630 200E6D ; JSR SETPH  
42 6633 B90706 ; LDA OBJTBL+7,Y ; SET BIT 4=1 FOR COLOR  
43 6636 0910 ; ORA #\$10  
44 6638 990706 ; STA OBJTBL+7,Y  
45 663B A905 ; LDA #5 ; RICOCHET SOUND  
46 663D 20A863 ; JSR SNDINI  
47 6640 4C7564 ; JMP COL160 ; NEXT  
48 ;  
49 ; ZORBA GOT IT  
50 ;  
51 6643 COL220  
52 ;  
53 ; <<<-->  
54 ;  
55 ; DO SOMETHING FANCY TO SHOW ZORBA BEAED  
56 ;  
57 ;

```

6645 A900
6647 2000EB  GLBPO  LDA   #$.0
6648 CA      RNE   AUDF1,X
6649 D05A 6647  RNE   GLBPO
664D A2B0  LDA   #$.B0
664F 85AC  STA   SYSTAT
6651 A980  LDA   #$.80
6653 8001E8  STA   AUDI1
6656 8005E8  STA   AUDI2
6659 A00A  B0QML  LDY   #$.A
665B A00B04  B00Z  LDA   VCOUNT
665E D0E9 6659  BNE   BOOML
6660 88    DEY
6661 D0F8 665B  BNE   B00Z
6663 A00AE8  LDA   RANDOM
6666 8D1209  STA   COLOR1
6669 A00AE8  LDA   RANDOM
666C 8D1309  STA   COLOR2
666F A00AE8  LDA   RANDOM
6672 8D1409  STA   COLOR3
6675 A00AE8  LDA   RANDOM
6678 8D1109  STA   COLOR0
667B A00AE8  LDA   RANDOM
667E 0908  ORA   #$.00001.000
6680 8D1509  STA   COLOR4
6683 A5AC  LDA   SYSTAT
6685 8D04E8  STA   AUDF3
6688 AA    TAX
6689 BD2775  LDA   ZORBD,X
668C 8D00E8  STA   AUDF1
668F C6AC  DEC   SYSTAT
6691 D0C6 6659  BNE   BOOML

```

; THIS PLAY LEVEL OVER - GO TO NEXT

```

6693 COL260
6693 A014  LDY   #$.20
6695 COL280  LDY   #$.10
6695 A210  LDY   #$.10
6697 A900  LDA   #$.0
6699 2089Z3  JSR   ADDSCR
669C 88    DEY
669D D0F6 6695  BNE   COL280
669F E6AE  INC   PLAYLV
66A1 4C8569  JMP   NEWLV

```

; GO TO NEXT LEVEL

; MOTHER ZORBA HIT TABLE CONTAINS X OFFSET TO CENTER FOR REACTOR BLAST

	MTRHTB	DB	18	; ZORBA SIZE 0
66A4 12		DB	18	; ZORBA SIZE 1
66A5 12		DB	18	; ZORBA SIZE 2
66A6 0E		DB	14	; ZORBA SIZE 3
66A7 0B		DB	11	; ZORBA SIZE 4
66A8 08		DB	8	; ZORBA SIZE 5
66A9 07		DB	7	; ZORBA SIZE 6
66AA 05		DB	5	; ZORBA SIZE 7
66AB 04		DB	4	; ZORBA SIZE 8
66AC 02		DB	2	

1 : ROUTINE CHECKS TO SEE IF BASE OBJECT (BUCK OR BUCK SHOT)  
2 : HAS COINCIDENCE W/ OBJECT INDEXED BY Y

3 : ENTER:

4 : Y=INDEX INTO OBJTBL TO TARGET

5 : \$10=BASE OBJECT LENGTH

6 : \$11=BASE OBJECT WIDTH

7 : \$12=BASE OBJECT X LOCATION

8 : \$13=BASE OBJECT Y LOCATION

9 : \$0B=TARGET OBJECT LENGTH

10 : \$0C=TARGET OBJECT WIDTH

11 : \$0D=DESTROYED

12 : \$0E=DESTROYED

13 : EXIT:

14 : X DESTROYED

15 : \$00-\$02 = USED

16 : \$1F-\$1E = Y SAVE

17 : \$0H-\$0F = USED

18 : IF A=00, NO COLLISION WAS DETECTED

19 : IF A=FF, COLLISION DETECTED

20 : CKCRSH

21 :  
22 :  
23 :  
24 :  
25 :  
26 :  
27 :  
28 :  
29 :  
30 :  
31 :  
32 :  
33 :  
34 :  
35 :  
36 :  
37 :  
38 :  
39 :  
40 :  
41 :  
42 :  
43 :  
44 :  
45 :  
46 :  
47 :  
48 :  
49 :  
50 :  
51 :  
52 :  
53 :  
54 :  
55 :  
56 :  
57 :  
58 :  
59 :  
60 :  
61 :  
62 :  
63 :  
64 :  
65 :  
66 :  
67 :  
68 :  
69 :  
70 :  
71 :  
72 :  
73 :  
74 :  
75 :  
76 :  
77 :  
78 :  
79 :  
80 :  
81 :  
82 :  
83 :  
84 :  
85 :  
86 :  
87 :  
88 :  
89 :  
90 :  
91 :  
92 :  
93 :  
94 :  
95 :  
96 :  
97 :  
98 :  
99 :  
100 :  
101 :  
102 :  
103 :  
104 :  
105 :  
106 :  
107 :  
108 :  
109 :  
110 :  
111 :  
112 :  
113 :  
114 :  
115 :  
116 :  
117 :  
118 :  
119 :  
120 :  
121 :  
122 :  
123 :  
124 :  
125 :  
126 :  
127 :  
128 :  
129 :  
130 :  
131 :  
132 :  
133 :  
134 :  
135 :  
136 :  
137 :  
138 :  
139 :  
140 :  
141 :  
142 :  
143 :  
144 :  
145 :  
146 :  
147 :  
148 :  
149 :  
150 :  
151 :  
152 :  
153 :  
154 :  
155 :  
156 :  
157 :  
158 :  
159 :  
160 :  
161 :  
162 :  
163 :  
164 :  
165 :  
166 :  
167 :  
168 :  
169 :  
170 :  
171 :  
172 :  
173 :  
174 :  
175 :  
176 :  
177 :  
178 :  
179 :  
180 :  
181 :  
182 :  
183 :  
184 :  
185 :  
186 :  
187 :  
188 :  
189 :  
190 :  
191 :  
192 :  
193 :  
194 :  
195 :  
196 :  
197 :  
198 :  
199 :  
200 :  
201 :  
202 :  
203 :  
204 :  
205 :  
206 :  
207 :  
208 :  
209 :  
210 :  
211 :  
212 :  
213 :  
214 :  
215 :  
216 :  
217 :  
218 :  
219 :  
220 :  
221 :  
222 :  
223 :  
224 :  
225 :  
226 :  
227 :  
228 :  
229 :  
230 :  
231 :  
232 :  
233 :  
234 :  
235 :  
236 :  
237 :  
238 :  
239 :  
240 :  
241 :  
242 :  
243 :  
244 :  
245 :  
246 :  
247 :  
248 :  
249 :  
250 :  
251 :  
252 :  
253 :  
254 :  
255 :  
256 :  
257 :  
258 :  
259 :  
260 :  
261 :  
262 :  
263 :  
264 :  
265 :  
266 :  
267 :  
268 :  
269 :  
270 :  
271 :  
272 :  
273 :  
274 :  
275 :  
276 :  
277 :  
278 :  
279 :  
280 :  
281 :  
282 :  
283 :  
284 :  
285 :  
286 :  
287 :  
288 :  
289 :  
290 :  
291 :  
292 :  
293 :  
294 :  
295 :  
296 :  
297 :  
298 :  
299 :  
300 :  
301 :  
302 :  
303 :  
304 :  
305 :  
306 :  
307 :  
308 :  
309 :  
310 :  
311 :  
312 :  
313 :  
314 :  
315 :  
316 :  
317 :  
318 :  
319 :  
320 :  
321 :  
322 :  
323 :  
324 :  
325 :  
326 :  
327 :  
328 :  
329 :  
330 :  
331 :  
332 :  
333 :  
334 :  
335 :  
336 :  
337 :  
338 :  
339 :  
340 :  
341 :  
342 :  
343 :  
344 :  
345 :  
346 :  
347 :  
348 :  
349 :  
350 :  
351 :  
352 :  
353 :  
354 :  
355 :  
356 :  
357 :  
358 :  
359 :  
360 :  
361 :  
362 :  
363 :  
364 :  
365 :  
366 :  
367 :  
368 :  
369 :  
370 :  
371 :  
372 :  
373 :  
374 :  
375 :  
376 :  
377 :  
378 :  
379 :  
380 :  
381 :  
382 :  
383 :  
384 :  
385 :  
386 :  
387 :  
388 :  
389 :  
390 :  
391 :  
392 :  
393 :  
394 :  
395 :  
396 :  
397 :  
398 :  
399 :  
400 :  
401 :  
402 :  
403 :  
404 :  
405 :  
406 :  
407 :  
408 :  
409 :  
410 :  
411 :  
412 :  
413 :  
414 :  
415 :  
416 :  
417 :  
418 :  
419 :  
420 :  
421 :  
422 :  
423 :  
424 :  
425 :  
426 :  
427 :  
428 :  
429 :  
430 :  
431 :  
432 :  
433 :  
434 :  
435 :  
436 :  
437 :  
438 :  
439 :  
440 :  
441 :  
442 :  
443 :  
444 :  
445 :  
446 :  
447 :  
448 :  
449 :  
450 :  
451 :  
452 :  
453 :  
454 :  
455 :  
456 :  
457 :  
458 :  
459 :  
460 :  
461 :  
462 :  
463 :  
464 :  
465 :  
466 :  
467 :  
468 :  
469 :  
470 :  
471 :  
472 :  
473 :  
474 :  
475 :  
476 :  
477 :  
478 :  
479 :  
480 :  
481 :  
482 :  
483 :  
484 :  
485 :  
486 :  
487 :  
488 :  
489 :  
490 :  
491 :  
492 :  
493 :  
494 :  
495 :  
496 :  
497 :  
498 :  
499 :  
500 :  
501 :  
502 :  
503 :  
504 :  
505 :  
506 :  
507 :  
508 :  
509 :  
510 :  
511 :  
512 :  
513 :  
514 :  
515 :  
516 :  
517 :  
518 :  
519 :  
520 :  
521 :  
522 :  
523 :  
524 :  
525 :  
526 :  
527 :  
528 :  
529 :  
530 :  
531 :  
532 :  
533 :  
534 :  
535 :  
536 :  
537 :  
538 :  
539 :  
540 :  
541 :  
542 :  
543 :  
544 :  
545 :  
546 :  
547 :  
548 :  
549 :  
550 :  
551 :  
552 :  
553 :  
554 :  
555 :  
556 :  
557 :  
558 :  
559 :  
560 :  
561 :  
562 :  
563 :  
564 :  
565 :  
566 :  
567 :  
568 :  
569 :  
570 :  
571 :  
572 :  
573 :  
574 :  
575 :  
576 :  
577 :  
578 :  
579 :  
580 :  
581 :  
582 :  
583 :  
584 :  
585 :  
586 :  
587 :  
588 :  
589 :  
590 :  
591 :  
592 :  
593 :  
594 :  
595 :  
596 :  
597 :  
598 :  
599 :  
600 :  
601 :  
602 :  
603 :  
604 :  
605 :  
606 :  
607 :  
608 :  
609 :  
610 :  
611 :  
612 :  
613 :  
614 :  
615 :  
616 :  
617 :  
618 :  
619 :  
620 :  
621 :  
622 :  
623 :  
624 :  
625 :  
626 :  
627 :  
628 :  
629 :  
630 :  
631 :  
632 :  
633 :  
634 :  
635 :  
636 :  
637 :  
638 :  
639 :  
640 :  
641 :  
642 :  
643 :  
644 :  
645 :  
646 :  
647 :  
648 :  
649 :  
650 :  
651 :  
652 :  
653 :  
654 :  
655 :  
656 :  
657 :  
658 :  
659 :  
660 :  
661 :  
662 :  
663 :  
664 :  
665 :  
666 :  
667 :  
668 :  
669 :  
670 :  
671 :  
672 :  
673 :  
674 :  
675 :  
676 :  
677 :  
678 :  
679 :  
680 :  
681 :  
682 :  
683 :  
684 :  
685 :  
686 :  
687 :  
688 :  
689 :  
690 :  
691 :  
692 :  
693 :  
694 :  
695 :  
696 :  
697 :  
698 :  
699 :  
700 :  
701 :  
702 :  
703 :  
704 :  
705 :  
706 :  
707 :  
708 :  
709 :  
710 :  
711 :  
712 :  
713 :  
714 :  
715 :  
716 :  
717 :  
718 :  
719 :  
720 :  
721 :  
722 :  
723 :  
724 :  
725 :  
726 :  
727 :  
728 :  
729 :  
730 :  
731 :  
732 :  
733 :  
734 :  
735 :  
736 :  
737 :  
738 :  
739 :  
740 :  
741 :  
742 :  
743 :  
744 :  
745 :  
746 :  
747 :  
748 :  
749 :  
750 :  
751 :  
752 :  
753 :  
754 :  
755 :  
756 :  
757 :  
758 :  
759 :  
760 :  
761 :  
762 :  
763 :  
764 :  
765 :  
766 :  
767 :  
768 :  
769 :  
770 :  
771 :  
772 :  
773 :  
774 :  
775 :  
776 :  
777 :  
778 :  
779 :  
770 :  
771 :  
772 :  
773 :  
774 :  
775 :  
776 :  
777 :  
778 :  
779 :  
780 :  
781 :  
782 :  
783 :  
784 :  
785 :  
786 :  
787 :  
788 :  
789 :  
790 :  
791 :  
792 :  
793 :  
794 :  
795 :  
796 :  
797 :  
798 :  
799 :  
800 :  
801 :  
802 :  
803 :  
804 :  
805 :  
806 :  
807 :  
808 :  
809 :  
8010 :  
8011 :  
8012 :  
8013 :  
8014 :  
8015 :  
8016 :  
8017 :  
8018 :  
8019 :  
8020 :  
8021 :  
8022 :  
8023 :  
8024 :  
8025 :  
8026 :  
8027 :  
8028 :  
8029 :  
8030 :  
8031 :  
8032 :  
8033 :  
8034 :  
8035 :  
8036 :  
8037 :  
8038 :  
8039 :  
8040 :  
8041 :  
8042 :  
8043 :  
8044 :  
8045 :  
8046 :  
8047 :  
8048 :  
8049 :  
8050 :  
8051 :  
8052 :  
8053 :  
8054 :  
8055 :  
8056 :  
8057 :  
8058 :  
8059 :  
8060 :  
8061 :  
8062 :  
8063 :  
8064 :  
8065 :  
8066 :  
8067 :  
8068 :  
8069 :  
8070 :  
8071 :  
8072 :  
8073 :  
8074 :  
8075 :  
8076 :  
8077 :  
8078 :  
8079 :  
8080 :  
8081 :  
8082 :  
8083 :  
8084 :  
8085 :  
8086 :  
8087 :  
8088 :  
8089 :  
8090 :  
8091 :  
8092 :  
8093 :  
8094 :  
8095 :  
8096 :  
8097 :  
8098 :  
8099 :  
80100 :  
80101 :  
80102 :  
80103 :  
80104 :  
80105 :  
80106 :  
80107 :  
80108 :  
80109 :  
80110 :  
80111 :  
80112 :  
80113 :  
80114 :  
80115 :  
80116 :  
80117 :  
80118 :  
80119 :  
80120 :  
80121 :  
80122 :  
80123 :  
80124 :  
80125 :  
80126 :  
80127 :  
80128 :  
80129 :  
80130 :  
80131 :  
80132 :  
80133 :  
80134 :  
80135 :  
80136 :  
80137 :  
80138 :  
80139 :  
80140 :  
80141 :  
80142 :  
80143 :  
80144 :  
80145 :  
80146 :  
80147 :  
80148 :  
80149 :  
80150 :  
80151 :  
80152 :  
80153 :  
80154 :  
80155 :  
80156 :  
80157 :  
80158 :  
80159 :  
80160 :  
80161 :  
80162 :  
80163 :  
80164 :  
80165 :  
80166 :  
80167 :  
80168 :  
80169 :  
80170 :  
80171 :  
80172 :  
80173 :  
80174 :  
80175 :  
80176 :  
80177 :  
80178 :  
80179 :  
80180 :  
80181 :  
80182 :  
80183 :  
80184 :  
80185 :  
80186 :  
80187 :  
80188 :  
80189 :  
80190 :  
80191 :  
80192 :  
80193 :  
80194 :  
80195 :  
80196 :  
80197 :  
80198 :  
80199 :  
80200 :  
80201 :  
80202 :  
80203 :  
80204 :  
80205 :  
80206 :  
80207 :  
80208 :  
80209 :  
80210 :  
80211 :  
80212 :  
80213 :  
802

## COINCIDENCE CHECKER

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 27  
J:5200 .A65

66EA

CKCB50

LPS  
RTS

#0

NO COLLISION

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

2 ; ROUTINE INITIALIZES POLE OBJECT DESCRIPTORS AT PERIODIC INTERVALS  
3

```

4 66E8 POLES
5 66E9 A5AC LDA SYSTAT ; SEE IF POLES THIS ROUND
6 66E9 2902 AND #2
7 66F1 D01B ^670E BNE POLE10 ; NO
8 66F3 A582 LDA POLCNT ; SEE IF OK TO PLACE POLES
9 66F5 D018 ^670F BNE POLE20 ; NO
10 66E7 A201 LDX #1 ; SEE IF ROOM FOR 2 POLES
11 66F9 20E06C JSR CREATE
12 66FC D010 ^670E BNE POLE10 ; NO ROOM
13 66FF 8400 STY 0 ; SAVE POINTER TO 1ST
14 6700 A201 LDX #1
15 6702 20E06C JSR CREATE
16 6703 F00D ^6714 BEQ POLE30 ; GOT 'EM
17 6707 A400 LDY 0 ; NO ROOM FOR 2ND = KILL FIRST
18 6709 A900 LDA #0
19 670B 980006 STA DR.ITBL,Y
20 670E POLE10 RTS
21 670E 60 POLE20 RTS
22 670F C482 DEC POLCNT ; DEC COUNTER
23 6711 4C0E67 JMP POLE10
24
25 ; GOT BOTH OBJECT DESCRIPTOR BASES - SET THEM UP
26
27 6714 POLE30
28 6714 8401 STY 1 ; SAVE 2ND DESCRIPTOR INDEX
29 6716 POLE35
30 6716 POLE40
31 6716 A80A58 LDA RANDOM ; GET RANDOM X
32 6719 E9B4 CMP #180 ; RIGHT POLE MUST BE LESS THAN 150
33 671B 80E9 ^6716 BCS POLE40
34 671D A401 LDY 1
35 671F 990106 STA DR.ITBL+1,Y ; SET RIGHT X (Y ALREADY 0)
36 6722 A400 LDY 0
37 6724 88 SEC
38 6725 E5A8 SBC SPREAD ; DISTANCE BETWEEN POLES
39 6727 90E9 ^6714 BCC POLE40 ; CAN'T PUT IT HERE
40 6729 C928 CMP #40 ; LEFT MUST BE >50
41 672B 20E9 ^6716 BCC POLE40 ; TRY AGAIN
42 672D 990106 STA DR.ITBL+1,Y
43 6730 A908 LDA #8 ; SET AS LEFT POLE
44 6732 990706 STA OBJTBL+7,Y
45 6735 A909 LDA #9 ; SET SIZE=9
46 6737 990306 STA DR.ITBL+3,Y
47 673A A401 LDY 1
48 673C 990306 STA DR.ITBL+3,Y
49 673E A920 LDA #20 ; SPEED ALWAYS SAME TO START
50 6741 990506 STA OBJTBL+5,Y
51 6744 A400 LDY 0
52 6746 990506 STA OBJTBL+5,Y
53 6749 A901 LDA #1 ; SET LEFT POLE AS PATH1
54 674B 200E6D JSR SETPTH
55 674E A401 LDY 1 ; SET RIGHT POLE AS PATH2
56 6750 A902 LDA #2
57 6752 200E6D JSR SETPTH
58 6755 A5A0 LDA POLDLY
59 6757 8582 STA POLCNT ; RESET DELAY COUNTER

```

6755 A5A0  
6757 6582

LDA  
STA

POLDLY  
POLCNT

RESET DELAY COUNTER

POLE PROCESSING

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 29  
J:5200 A65

6759 60

RTS

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

## 675A PAINT

```

1 675A A51D    LDA    #$10
2 675C 3B       SEC
3 675D E720    SBC    #32
4 675F A8       TAY
5 6760 A61E    LDX    $1E
6 6762 8479    STX    YCOOD
7 6764 B9157D    LDA    Q4,Y
8 6767 8570    STA    BYTE
9 6769 B9157E    LDA    R4,Y
10 676C 856E    STA    BIT
11 676E A8       TAY
12 676F B16B    LDA    TADDR1,Y
13 6771 8576    STA    SHARE
14 6773 B16D    LDA    TADDRM1,Y
15 6775 8572    STA    MASK
16 6777 C8       INY
17 6778 B16B    LDA    TADDR1,Y
18 677A 8577    STA    SHARE+1
19 677C B16D    LDA    TADDRM1,Y
20 677E 8578    STA    MASK+1
21 6780 A008    LDY    #$8
22 6782 B16B    LDA    TADDR1,Y
23 6784 A200    LDX    #$0
24 ; STA    TADDRB,X1
25 6786 A679    LDX    YCOOD
26 6788 857F    STA    WIDTH
27 678A 18       CLC
28 678B 2470    BIT    BYTE
29 678D 102B    ^678A    BPL    RCLIP
30 678F 6570    ADC    BYTE
31 6791 9024    ^67B9    BCC    EXITLC
32 6793 F024    ^67B9    BEQ    EXITLG
33 6795 857E    STA    WIDTHC
34 6797 A57E    LDA    WIDTH
35 6799 38       SEC
36 679A E57E    SBC    WIDTHC
37 679C 18       CLC
38 679D 6576    ADC    SHARE
39 679F 8576    STA    SHARE
40 67A1 9002    ^67A5    BCC    GL1
41 67A3 E677    INC    SHARE+1
42 67A5 A57F    GL1    LDA    WIDTH
43 67A7 38       SEC
44 67A8 E57E    SBC    WIDTHC
45 67AA 18       CLC
46 67AB 6572    ADC    MASK
47 67AD 8572    STA    MASK
48 67AF 9002    ^67B3    BCC    GL2
49 67B1 E673    INC    MASK+1
50 67B3 A900    GL2    LDA    #$0
51 67B5 8570    STA    BYTE
52 67B7 F00F    ^67C8    BEQ    A1
53 67B9 60       EXITLC    RTS
54 67BA 857F    RCLIP    STA    WIDTHC
55 67BC 6570    ADC    BYTE
56 67BE C928    CMP    #40
57 67C0 9006    ^67C8    BCC    A1
58 67C2 A928    LDA    #40
59 67C4 E570    SBC    BYTE

```

## PAINT ROUTINE

```

1 67C6 857F
2 67C8 C8 A1 STA WIDTHC
3 67C9 B14B LDA [ADDR1],Y
4 67CB 8574 STA HIGHT
5 67CD A001 LDY #$1
6 ; STA [ADDRB1],Y
7 ; LDA ADDR
8 67CF 18
9 67D0 6902 ; LDA
10 ; ADC #$2
11 ; STA BACK
12 67D2 6900 ; LDA ADDR+1
13 ; ADC #$0
14 ; STA BACK+1
15 67D4 E090 HERE LDX #144
16 67D6 B01F ^67F7 BCS TH
17 67D8 B0B17E MDA PTH,X
18 67DB 8575 STA SCREEN+1
19 67DD B0157E LDA P1L,X
20 67E0 18 CLC
21 67E1 6570 ADC BYTE
22 67E2 8574 STA SCREEN
23 67E5 9002 ^67E9 BCC OKP1
24 67E7 E675 INC SCREEN+1
25 67E9 A47E OKP1 LDY WIDTHC
26 67FB 88 DEY
27 67FC R174 DRAW LDA [SCREEN],Y
28 67FE 9172 AND [MASK],Y ;<----BUG3
29 67F0 1176 BAA [SHAPE1],Y ;<----nun
30 67F2 9174 STA [SCREEN1],Y
31 67F4 88 DEY
32 67F5 10E5 ^67EC BPL DRAW
33 67F7 18 TH CLC
34 67F8 A57F LDA WIDTH
35 67FA 6572 ADC MASK
36 67EC 8572 STA MASK
37 67FF 9002 ^6802 BCC GL3
38 6800 E673 INC MASK+1
39 6802 A57F GL3 LDA WIDTH
40 6804 18 CLC
41 ; ADC BACK
42 ; STA BACK
43 ; BCC GL4
44 ; INC BACK+1
45 6807 A57F GL4 LDA WIDTH
46 6809 18 CLC
47 680A 6574 ADC SHAPE
48 680C 8576 STA SHAPE
49 680E 9002 ^6812 BCC GL5
50 6810 E677 INC SHAPE+1
51 6812 E8 GL5 INX
52 6813 C671 DEC HIGHT
53 6815 B0BD ^67D4 BNE HERE
54 6817 60 RTS

```

1: ROUTINE ERASES AN OBJECT FROM VIDRAM

2: ENTER:

3: 1C = PICTURE NUMBER TO DRAW

4: 1D = X DISPLAY POSITION

5: 1E = Y DISPLAY POSITION

6: 1F = FG COLOR

7: .00-01 = USED

8: .05-07 = USED

9: 10-1B = USED

10: 1C-1C = CHANGED

11: ;

12: ;

13: ;

16: 6818	ERASE		
17: 6818	A51D	LDA \$1D	
18: 681A	38	SEC	
19: 681B	E920	SBC #32	
20: 681D	A8	TAY	
21: 681E	A61E	LTX \$1E	
22: 6820	8679	STX YC00D	
23: 6822	B9157D	LDA Q4,Y	
24: 6825	8570	STA BYTE	
25: 6827	B9157E	LDA R4,Y	
26: 682A	856F	STA BIT	
27: 682C	A8	TAY	
28: 682D	B16B	LDA [ADDR1],Y	
29: 682E	8576	STA SHAPE	
30: 6831	B14D	LDA [ADDRM1],Y	
31: 6833	8572	STA MASK	
32: 6835	C8	INY	
33: 6836	B16B	LDA [ADDR1],Y	
34: 6838	8577	STA SHAPE+1	
35: 683A	B16D	LDA [ADDRM1],Y	
36: 683C	8573	STA MASK+1	
37: 683E	A008	LDY #\$8	LOCATION OF SHAPE SIZE
38: 6840	B16B	LDA [ADDR1],Y	
39: 6842	A200	LDX #\$0	
40: 6843	38	STA [ADDRB],XL	
41: 6844	A679	LDX YC00D	
42: 6846	857F	STA WIDTH	
43: 6848	18	CLC	
44: 6849	2470	BIT BYTE	
45: 684B	102B ^6878	BPL ZRCLIP	
46: 684D	6570	ADC BYTE	
47: 684E	9026 ^6877	BCC ZEXITLC	
48: 6851	F024 ^6877	BEQ ZEXITLC	
49: 6853	857E	STA WIDTHC	
50: 6855	A57F	LDA WIDTH	
51: 6857	38	SEC	
52: 6858	E57E	SBC WIDTHC	
53: 685A	18	CLC	
54: 685B	6576	ADC SHAPE	
55: 685D	8576	STA SHAPE	
56: 685F	9002 ^6863	BCC 7RI 1	
57: 6861	F677	INC SHAPE+1	
58: 6863	A57F	ZGL1 LDA WIDTH	
59: 6865	38	SEC	

6861 F477 TNC SHAPE+1  
6863 A57F ZGL1 LDA WIDTH  
6865 38 SEC

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 33  
ERASE OBJECT ROUTINE J:5200 .A65

6866 E57E SBC WIDTHC  
6868 18 CLC  
6869 6572 ADC MASK  
686B 8572 STA MASK  
686D 9002 ^6871 BCC ZGL2  
686E E673 INC MASK+1  
6871 A900 LDA #\$0  
6873 8570 STA BYTE  
6875 F00F ^6886 BEQ ZA1  
6877 60 ZEXITLC RTS  
6878 857E ZRCLIP STA WIDTHC  
687A 6570 ADC BYTE  
687C C928 CMP #40  
687E 9006 ^6886 BCC ZA1  
6880 A928 LDA #40  
6882 E570 SBC BYTE  
6884 857E STA WIDTHC  
6886 C8 ZA1 LDY  
6887 B16B LDA [ADDR1,Y]  
6889 8571 STA HEIGHT  
688B A579 LDA YC000  
688D 4A LSR A  
688E 18 CLC  
688F 690A ADC #10  
6891 C970 CMP #\$70  
6893 9002 ^6897 BCC ERGTIT  
6895 A970 LDA #\$70  
6897 C00BD4 ERGTIT CMP VCOUNT  
689A B0FB ^6897 BCS ERGTIT  
689C A001 LDY #\$1  
689E 18 CLC  
689F 6902 ADC #\$2  
68A1 6900 ADC #\$0  
68A3 E020 ZHERE CBX #144  
68A5 B01D ^68C4 BCS ZTH  
68A7 BDB17E LDA P1H,X  
68AA 8575 STA SCREEN+1  
68AC BD157E LDA P1L,X  
68AF 18 CLC  
68B0 6570 ADC BYTE  
68B2 8574 STA SCREEN  
68B4 9002 ^68B8 BCC ZOKP1  
68B6 E475 INC SCREEN+1  
68B8 A47E ZOKP1 LDY WIDTHC  
68B9 88 DEY  
68BB B174 ZDRAW LDA [SCREEN],Y  
68BD 3172 AND [MASK1,Y] :<----BUG?  
68BE 9174 STA [SCREEN],Y  
68C1 88 DEY  
68C2 10F7 ^68BB BPL ZDRAW  
68C4 18 ZTH CLC  
68C5 A57F LDA WIDTH  
68C7 6572 ADC MASK  
68C9 8572 STA MASK  
68CB 9002 ^68CF BCC ZGL3  
68CD E673 INC MASK+1  
68CF A57F ZGL3 LDA WIDTH  
68D1 18 CLC  
; ADC BACK  
; STA BACK

## ERASE OBJECT ROUTINE

68D2	9000	^68D4	BCC	ZGL4
68D4	A57F	ZGL4	INC	BACK+1
68D6	18		LDA	WIDTH
68D7	6576		CLC	
68D9	8576		ADC	SHAPE
68D8	9002	^68DE	STA	SHAPE
68D9	9002	^68DE	BCC	ZGL5
68DD	E677		INC	SHAPE+1
68DF	E8	ZGL5	INX	
68E0	C671		DEC	HIGHI
68E2	DQBF	^68A3	BNE	ZHERE
68E4	60		RTS	

1 : CONTROL REGISTERS INITIALIZED AS FOLLOWS:

2  
3  
4  
5  
6 68E5 CINIT  
7 68E5 78 SEI  
8 68E6 D8 CLD  
9 68E7 A900 LDA #\$00  
10 68E9 8D0E14 STA NMIFEN ; DON'T BE RUDE!! (AND INTERRUPT)  
11 68EC AA TAX  
12 68ED 7A PIT  
13 68E9 9D00C0 STA \$0000,X  
14 68F0 9D00F4 STA \$D400,X  
15 68F3 9D00E8 STA \$E800,X  
16 68F6 9D00D3 STA \$D300,X  
17 68F9 9D0006 STA \$600,X  
18 68FC 9D0002 STA \$200,X  
19 68FF 9500 STA \$00,X  
20 6901 E8 INX  
21 6902 D0E9 ^68ED BNE ZAPIT  
22  
23 : ENTRANCE FOR RESTART  
24  
25 6904 INIT  
26 6904 A940 LDA #Z01000000  
27 6906 8D0E08 STA \$E80E  
28 6909 A900 LDA #\$0  
29 690B 8D00D4 STA \$D400  
30 690E 8D10C0 STA GRACTL  
31 6911 A9B1 LDA #LOW DLILOC ; INIT DLI VECTOR  
32 6913 8D0602 STA VDSLST  
33 6914 A900 LDA #HIGH DLILOC  
34 6918 8D0702 STA VDSLST+1  
35 691B A9DE LDA #LOW VRIRTN ; INIT VRI VECTOR  
36 691D 8D0202 STA VRIRKT  
37 6920 A973 LDA #HIGH VRIRTN  
38 6922 8D0302 STA VRIRKT+1  
39 6925 A902 LDA #Z00000010 ; INVERSE CHARS  
40 6927 8D0409 STA CHART  
41 692A A93C LDA #Z00111100 ; SETUP PIA FOR JOYSTICK READ  
42 692C 8D0203 STA FACTL  
43 692F 8D0303 STA PBCTL  
44 6932 A902 LDA #Z00000010 ; SETUP PUKEY FOR KEYBOARD READ AND DEBOUNCE  
45 6934 8D0F08 STA \$E80E  
46 6937 8D0FE8 STA SKCTL  
47 693A 206573 JSR CHKHSC ; CHECK HIGH SCORE  
48 693D 206371 JSR TITLE ; DISPLAY TITLE PAGE  
49 6940 A997 LDA #HIGH MOUDA ; MUST BE ON A 2K BOUNDARY  
50 6942 8D0309 STA CHBAS  
51 6945 A900 LDA #Z01000000  
52 6947 8D0E04 STA NMIFEN  
53 694A A941 LDA #LOW DLILST ; INIT DISPLAY LIST (CANNOT CROSS 1K BOUND.)  
54 694C 8D0002 STA SDLSTL  
55 694F A980 LDA #HIGH DLILST  
56 6951 8D0109 STA SDLSTH  
57  
58 ; COPY DLILST INTO ZERO PAGE  
59 ;

```

1 6954 A240
2 6954 BDEB80 CLOOP LDX #$40
3 6959 95B1 STA BEFLOC,X
4 695B CA DEX BLILOC,X
5 695C 10F8 ^6956 BPL CLOOP
6 695E A902 LDA #$2
7 6960 857D STA SPEEDC
8 6962 8DCA00 STA SPEED
9
10 6965 A905 LDA #5 ; FIRE DELAY AND FUDGE AMOUNT
11 6967 85AF STA FIRDLY
12 6969 85AD STA SHPLFT
13 696B A908 LDA #8
14 696D 85B0 STA FIRFDG

```

; INITIALIZE VARTABLES THAT ARE NONE ONLY AT BEGINNING OF GAME.  
; DATA DERIVED FROM TABLE

```

18 696F A900 LDA #0 ; STARTING PLAY LEVEL =0
19 6971 857B STA LYFLG
20 6973 85AC STA SYSTAT
21 6975 85AF STA PLYLVL
22 6977 857C STA MEANFG
23 6979 8D0009 STA CSCORE ; ZERO OUT SCORE
24 697C 8D0F09 STA CSCORE+1
25 697F 8D0F09 STA CSCORE+2
26 6982 8D1009 STA CSCORE+3

```

; DATA TO BE INITIALIZED BEFORE EACH PLAY LEVEL

```

31 6985 NEWLVL
32 6985 A5AE LDA PLYLVL ; GET CURRENT PLAY LEVEL
33 6987 F914 CMP #MAXLVL ; CHECK FOR MAXIMUM LEVEL
34 6989 9000 ^6997 BCC NEWL05
35 698B 38 SEC ; IF PAST MAXIMUM, GO BACK 4
36 698C F904 SBC #4
37 698E 85AE STA PLYLVL ; REPEAT LAST 4 FOREVER
38 6990 A57C LDA MEANFG
39 6992 18 CLC
40 6993 6905 ADC #5
41 6995 857C STA MEANFG
42 6997 NEWL05
43 U 6997 200000 JSR LEVELD
44 699A A5AE LDA PLYLVL
45 699C A20F LDY #PTENT ; *NUMBER OF DATA BYTES TO INITIALIZE
46 699E 20C273 JSR MULT
47 69A1 8501 STA 1
48 69A3 8600 STX 0
49 69A5 A925 LDA #LOW PLYTBL ; OFFSET TO BASE OF TABLE
50 69A7 18 CLC
51 69A8 6500 ADC 0
52 69A9 8500 STA 0
53 69AC A96A LDA #HIGH PLYTBL
54 69AE A501 ADC 1
55 69B0 8501 STA 1
56 69B2 A000 LDY #0 ; INIT INDEX
57 69B4 NEWL10
58 69B4 B100 LDA [0],Y
59 69B6 999FOO STA LVLDAT,Y

```

69B9 C8  
69BA 000E  
69BC D0E6 ^69B4

INY  
CFY  
BNE #RTCNT  
NEWL10

; DON'T LET THEM PLAY FOREVER

69BE A5A7  
69C0 18  
69C1 657C  
69C3 85A7

LDA MINSPO  
CLC  
ADC MEANFG  
STA MINSPO

; SEE IF TO GIVE NEW TANK OF FUEL

69C5 A5AE  
69C7 2903  
69C9 D00A ^69D5  
69CB A960  
69CD 859E  
69CF A900  
69D1 859C  
69D3 859D  
69D5

LDA PLYLVL  
AND #3  
BNE NEWL20  
LDA #96  
STA FILAMT+2  
LDA #0  
STA FILAMT  
STA FILAMT+1

; ONLY GIVE NEW FUEL WHEN STARTING 1ST OF 4

; NEW TANK - START WITH AMOUNT DISPLAYABLE

NEWL20

; DATA TO BE INITIALIZED BEFORE EACH PLAY

69D5  
69D5 A92F  
69D7 8597  
69D9 A900  
69D8 8598

LDA #47  
STA MISIRT  
LDA #0  
STA MTNCNT

; INITIALIZE BACKDROP

69D0 202560  
69E0 209960  
69F3 A5AF

JSR BDINIT  
JSR SETCOL  
LDA BYSTAT  
AND #%00000010

69E5 2902  
69E7 D003 ^69FD  
69F9 208260

RNE SKPDLI  
JSR SETDLI

; SET BUCK START LOCATION

69EC A960  
69EB 859B  
69F0 A570

LDA #26  
STA BUCKY  
LDA #11

; CENTER

69E2 859A  
69F4 20B262  
69F7 A5A7

JSR SHIPS  
LDA MINSPO

; FORCE INITIAL SHIP DISPLAY

69F9 B599  
69FB 206A60  
69FF A900

STA SCRSPD  
JSR CONVSP  
LDA #0

6A00 859A  
6A02 8D01E8  
6A05 8D03E8

STA MTNSPO  
STA AUDC1

;

6A08 8D05E8  
6A0B 8D07E8  
6A0F 8589

STA AUDC2  
STA AUDC3  
STA AUDC4

; TURN ON SOUND GENERATOR

6A10 A8  
6A11 NEWP10

STA SDSTAT  
STA TAY

;

6A11 A900  
6A13 990006  
6A16 C8

LDA #0  
STA OBJTBL,Y  
INY

;

6A17 D0E8 ^6A11

BNE NEWP10

;

1 ---> DISPLAY INITIAL FUEL GRAPH (JSR FUEL10) AND INITIAL  
2 UFODSP (JSR UFODSP)

3 6A19 200163 JSR FUEL10  
4 6A1C 208C60 JSR UFODSP  
5 6A1F A2FF LDX #\$FF ; INIT STACK  
6 6A21 9A TXS  
7 6A22 402B60 JMP DRIVER ; DONE - NOW RUN PROGRAM  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

1 : PLAY LEVELS TABLE. CONTAINS THE DIFFICULTY PARAMETERS FOR THE  
 2 : VARIOUS LEVELS OF PLAY

3 : BYTES ARE ORDERED AS FOLLOWS:

4 :  
 5 : RSPONS JOY STICK RESPONSE  
 6 : PULDLY NUMBER OF PASSES BETWEEN POLE SETS  
 7 : SAUDLY NUMBER OF PASSES BETWEEN SAUCER SETS  
 8 : SAUSPD BASE SPEED OF SAUCERS  
 9 :  
 10 : HOPDLY NUMBER OF PASSES BETWEEN HOPPER APPEARANCES  
 11 : HOPSPD BASE SPEED OF HOPPERS  
 12 : MTRSPD MOTHER ZORBA SPEED  
 13 :  
 14 : UFDCNT NUMBER OF UFDOS (ROLES) TO KILL  
 15 : MINSPD MINIMUM SPEED BUCK CAN TRAVEL  
 16 : SPREAD DISTANCE BETWEEN POLES  
 17 : HERDLY HOPPER FIRE DELAY  
 18 : PERDLY POLE FIRE DELAY  
 19 : MERDLY MOTHER ZORBA FIRE DELAY  
 20 : SYSTAT SYSTEM PLAY LEVEL STATUS BITS  
 21 : 7 = HOPPERS FIRE  
 22 : 6 = POLES FIRE IN  
 23 : 5 = POLES FIRE OUT  
 24 : 4 = HOPPERS EXIST  
 25 : 3 = SAUCERS EXIST  
 26 : 2 = MOTHER ZORBA  
 27 : 1 = SPACE SCENE  
 28 : 0 =

29 6A25 PLYTBL

30 RSP PDL SDL SSP HDL HSP MSP UFD MIN SPR HFD FFD MED SYS

31 6A25 0420000000 DB \$04, \$20, \$00, \$00, \$00, \$00, \$0A, \$35, \$25, \$00, \$00, \$00, \$00

32 6A33 041E583000 DB \$04, \$1E, \$58, \$30, \$00, \$00, \$0F, \$35, \$25, \$00, \$00, \$00, \$08

33 6A41 041D583040 DB \$04, \$1D, \$58, \$30, \$40, \$20, \$00, \$14, \$35, \$25, \$00, \$00, \$18

34 6A4F 0400203000 DB \$04, \$00, \$20, \$30, \$00, \$00, \$60, \$0A, \$35, \$00, \$00, \$00, \$08, \$0A

35 6A50 0414000000 DB \$04, \$14, \$00, \$00, \$00, \$00, \$0F, \$3C, \$23, \$00, \$20, \$00, \$20

36 6A6B 0414503800 DB \$04, \$14, \$50, \$38, \$00, \$00, \$14, \$3C, \$23, \$00, \$28, \$00, \$28

37 6A79 0414503835 DB \$04, \$14, \$50, \$38, \$35, \$30, \$00, \$19, \$3C, \$23, \$38, \$28, \$00, \$B8

38 6A87 0400203800 DB \$04, \$00, \$20, \$38, \$00, \$00, \$60, \$0F, \$40, \$00, \$00, \$06, \$0A

39 6A95 0410000000 DB \$04, \$10, \$00, \$00, \$00, \$00, \$14, \$48, \$20, \$00, \$18, \$00, \$20

40 6AA3 0412403C20 DB \$04, \$12, \$40, \$3C, \$20, \$20, \$00, \$19, \$48, \$23, \$00, \$20, \$00, \$38

41 6AB1 0412403C20 DB \$04, \$12, \$40, \$3C, \$20, \$30, \$00, \$1E, \$48, \$23, \$28, \$20, \$00, \$B8

42 6ABF 0400203800 DB \$04, \$00, \$20, \$3C, \$00, \$00, \$60, \$14, \$50, \$00, \$00, \$00, \$04, \$0A

43 6ACD 040C803000 DB \$04, \$0C, \$80, \$30, \$00, \$00, \$19, \$50, \$20, \$20, \$10, \$00, \$28

44 6AD8 040E20301A DB \$04, \$0E, \$20, \$30, \$1A, \$40, \$00, \$1F, \$50, \$20, \$18, \$10, \$00, \$B8

45 6AE9 040E20301A DB \$04, \$0E, \$20, \$30, \$1A, \$40, \$00, \$28, \$50, \$20, \$10, \$00, \$98

46 6AF7 0400203000 DB \$04, \$00, \$20, \$30, \$00, \$00, \$60, \$20, \$58, \$00, \$00, \$00, \$04, \$0A

47 6B05 040A605000 DB \$04, \$0A, \$60, \$50, \$00, \$00, \$1E, \$70, \$14, \$00, \$08, \$00, \$28

48 6B13 040B185018 DB \$04, \$0B, \$18, \$50, \$18, \$40, \$00, \$23, \$60, \$18, \$10, \$08, \$00, \$B8

49 6B21 040B185010 DB \$04, \$0B, \$18, \$50, \$10, \$50, \$00, \$28, \$60, \$18, \$08, \$04, \$00, \$28

50 6B2F 0400185000 DB \$04, \$00, \$18, \$50, \$00, \$00, \$70, \$25, \$58, \$00, \$00, \$00, \$04, \$0A

51 0014 MAXLVI EQU (\*-PLYTBL)/PICNT

1 ROUTINE MAINTAINS VARIABLES ASSOCIATED WITH SCROLLING  
2 MOUNTAIN RANGE AND CAUSES MOUNTAIN RANGE TO SCROLL

```

5 6B3D MTNWRK
6 6B3D A59F LDA RSPONS ; GET SLOWDOWN AMOUNT
7 6B3F 4A LSR A
8 6B40 D002 ^6B44 BNE MTNW05
9 AR42 A901 LDA #1 ; CANT BE 0
10 6B44 MTNW05
11 6B44 8500 STA 0
12 6B46 A560 LDA JSDATA ; SEE IF GOING RIGHT
13 6B48 101C ^6B66 BPL MTNW30 ; YES
14 6B4A 2910 AND #$10 ; LEFT?
15 6B4C F024 ^6B72 BEQ MTNW40 ; YES
16
17 6B4E A596 LDA MTNSPD ; JOYSTICK UP - SLOW DOWN MOUNTAINS
18 6B50 100A ^6B5C BPL MTNW10 ; POSITIVE MTNSPD
19 6B52 18 CLC ; ADD TO SLOW DOWN NEGATIVE SPEED
20 6B53 6500 ADC 0
21 6B55 3024 ^6B7B BMI MTNW50
22 6B57 A200 LDA #0 ; STOP AT 0
23 6B59 4C7B6B JMP MTNW50
24 6B5C MTNW10
25 6B5C 38 SEC ; SUBTRACT TO SLOW DOWN POSITIVE MTNSPD
26 6B5D E500 SBC 0
27 6B5F 101A ^6B7B BPL MTNW50
28 6B61 A200 LDA #0 ; STOP AT 0
29 6B63 4C7B6B JMP MTNW50
30
31 6B66 MTNW30
32 6B66 A596 LDA MTNSPD ; JOYSTICK RIGHT - DEC MTNSPD
33 6B68 38 SEC
34 6B69 E59F SBC RSPONS
35 6B6B 500E ^6B7B BVC MTNW50
36 6B6D A980 LDA #-128 ; MIN =-128
37 6B6F 4C7B6B JMP MTNW50
38 6B72 MTNW40
39 6B72 A596 LDA MTNSPD ; JOYSTICK LEFT - INC MTNSPD
40 6B74 18 CLC
41 6B75 659F ADC RSPONS
42 6B77 5002 ^6B7B BVC MTNW50
43 6B79 A97F LDA #127 ; MAX=127
44 6B7B MTNW50 STA MTNSPD
45 6B7B 8596
46 ; SET UP TO DRAW MOUNTAINS
47 ;
48
49 6B7D MTNW60
50 6B7D 18 CLC
51 6B7F A596 LDA MTNSPD ; GET ABS(MTNSPD)
52 6B80 1005 ^6B87 RPL MTNW20
53 6B82 49FF EOR #$FF ; 2's COMP
54 6B84 6901 ADC #1
55 6B86 18 CLC
56 6B87 MTNW70
57 6B87 4A LSR A ; /2
58 6B88 4A LSR A ; /4
59 6B89 6598 ADC MTNCNT ; ADD TO CURRENT COUNT BASE

```

488B 8598	STA	MTNCNT	
6B8B 1080 ^6BCC	BPL	MTN130	: DONT MOVE MOUNTAINS
6B8F 227F	AND	###7F	: RESET TOP BIT FOR NEXT TIME
6B91 8598	STA	MTNCNT	
6B93 A596	LDA	MTNSPD	: SEE IF TO MOVE LEFT OR RIGHT
6B95 100F ^6BA6	BPL	MTNW80	: POSITIVE SPEED - DEC MISTRT
6B97 E697	INC	MTSTRT	: INC MOUNTAIN START ADDRESS
6B99 A597	LDA	MTSTRT	: CHECK ROLLOVER
6B9B C94E	CMP	#78	
6B9D D013 ^6BB2	BNE	MTNW90	: DRAW NEW MOUNTAIN RANGE
6B9F A321	LDA	#33	
6BA1 8597	STA	MTSTRT	
6BA3 4CB2AB	IMP	MTNW90	

6BA6	MTNW80		
6BA6 D697	DEC	MTSTRT	
6BA8 A597	DA	MTSTRT	
6BAA C920	CMP	#32	
6BAC D004 ^6BB2	BNE	MTNW90	: DRAW NEW MOUNTAIN RANGE
6BAE A94D	LDA	#77	
6BA0 8597	STA	MTSTRT	

: DRAW MOUNTAIN RANGE #0260-#0277

6BB2	MTNW90		
6BB2 A5AC	LDA	SYSTAT	
6BB4 2902	AND	#%00000010	
6BB6 D014 ^6BCC	BNE	MTN130	
6BB8 A000	LDY	#00	
6BBA A697	LDX	MTSTRT	

6BBC	MTNWAO		
6BBC 8A	TXA		
6BBD 29D820	STA	\$2008,Y	
6B00 E8	TNX		
6B01 E04E	CPX	#78	
6BC3 D002 ^6BC7	BNE	MTOK	
6BC5 A221	LDX	#33	
6BC7 C8	MTOK	INY	
6BC9 E028	CPY	#40	
6BCA D0F0 ^6BBG	BNE	MTNWAO	
6BCC	MTN130		
6BCC 40	RTS		

; ROUTINE CREATES 1-4 SAUCERS (RANDOM) STARTING ON THE  
; SAME PATH BUT AT DIFFERENT DISTANCES INTO THE PATH.  
; WON'T CREATE UNLESS COUNT IS SET.

```

5 6BCD  SAUCER
6 6BCD A5AC  LDA    SYSTAT   ; SEE IF THEY EXISTS HERE
7 6BCF 2908  AND    #8
8 6BD1 F004 ^6BD9  BEQ    SAUC10 ; NO
9 6BD3 A583  LDA    SAUCNT
10 6BD5 F003 ^6BDA  BEQ    SAUC20 ; CREATE SAUCERS
11 6BD7 C483  DEC    SAUCNT ; JUST DEC COUNT AND EXIT
12 6BD9  SAUC10
13 6BD9 60    RTS
14 6BDA  SAUC20
15 6BDA A902  LDA    #2    ; SEE HOW MANY SAUCERS TO CREATE (1-4)
16 6BDC 851F  STA    $1F   ; SAVE COUNTER
17 6BDE A00A8  SAUX  LDA    RANDOM ; GET STARTING X
18 6BE1 297F  AND    #%01111111
19 6BE3 18    CLC
20 6BE4 6920  ADC    #32
21 6BE4 851F  STA    $1F
22 6BF3 A00A8  LDA    RANDOM ; GET PATH (3,4,5,6)
23 6BF3 2903  AND    #3
24 6BED 18    CLC
25 6BEE 6903  ADC    #3
26 6BF0 851D  STA    $1D
27 6BF2  SAUC30
28 6BF2 A202  LDX    #2    ; TRY TO CREATE A SAUCER
29 6BF4 20E040  JSR    CREATE
30 6BF7 D0E0 ^6BD9  BNE    SAUC10 ; NO MORE ROOM - EXIT
31 6BF9 A51D  LDA    $1D    ; SET PATH
32 6BFB 200E40  JSR    SETRTH
33 6BFE A51E  LDA    $1F    ; SET X
34 6C00 290106  STA    OBJTBL+1,Y
35 ;
36 ; SET INITIAL RPTCNT=OBJECT#*37 (FROM TABLE)
37 ; Y=RPTCNT+144
38 ;
39 6C03 A61F  LDX    $1F    ; GET OBJECT#
40 6C05 BD1F60  LDA    RPTTBL,X ; GET #*28
41 6C08 290406  STA    OBJTBL+6,Y ; SET RPTCNT
42 6C0B 18    CLC    ; SET Y
43 6C0C 6990  ADC    #144
44 6C0E 990206  STA    OBJTBL+2,Y
45 ;
46 6C11 A5A2  LDA    SAUSPD ; SET SAUCER SPEED
47 6C13 290506  STA    OBJTBL+5,Y
48 6C16 C61F  DEC    $1F
49 6C18 F0D8 ^6BF2  BPL    SAUC30
50 6C1A  SAUC40
51 6C1A A5A1  LDA    SAUDLY ; RESET COUNT
52 6C1C 8583  STA    SAUCNT
53 6C1E 60    RTS
54 6C1F  RPTTRI
55 6C1F 00254A6F  DB     0,37,74,111

```

1 : DOES NECESSARY WORK FOR PROCESSING BUCK ROGER SHIP  
 2 : WHEN JOYSTICK PULLED LEFT OR RIGHT, BS MOVES LEFT OR RIGHT  
 3 : UP TO BOUNDARIES. WHEN JOYSTICK PUSHED UP OR PULLED DOWN,  
 4 : BUCK SHIP MOVES UP OR DOWN UNTIL BOUNDARIES ARE HIT AT  
 5 : WHICH TIME SCRSPD IS INCREASED (TOP BOUNDARY) OR DECREASED  
 6 : (LOWER BOUNDARY). WHEN FIRE BUTTON PRESSED, SHOTS ARE FIRED  
 7 : STARTING FROM BS AND TAKING A SPECIAL PATH DEFINED FOR  
 8 : BUCK SHOTS.  
 9 :  
 10 : BUCK SHIP CAN GO NO LOWER THAN 128, AND NO HIGHER THAN 96.  
 11 : X LIMITS ARE 48<X<176 SHADOW COPIES BUCK X BUT Y IS FIXED  
 12 : AT 131.

13 :  
 14 : 6C23 BSWORK  
 15 : 6C23 203273 ISR BSERAS : ERASE BUCK ROGERS

16 :  
 17 : 6C26 A560 LDA JSDATA : MOVE BUCK  
 18 : 6C28 1010 ^6C3A BEI BSW030 : RIGHT  
 19 : 6C2A 2910 AND #\\$10  
 20 : 6C2C F01D ^6C4A BEQ BSW040 : LEFT  
 21 :  
 22 : 6C2E A901 LDA #1 : FORCE BUCK CENTER PICTURE  
 23 : 4C30 8580 STA LASTBS  
 24 : 4C32 4C5760 JMP BSW050 : SKIP CENTER FLOAT  
 25 : : LDA BUCKX : FLOAT BUCK TO CENTER  
 26 : : CMP #112 : SEE IF LEFT OR RIGHT OF CENTER  
 27 : : BEQ BSW050 : AT CENTER - JUST DRAW IT  
 28 : : BMI BSW010 : LEFT OF CENTER  
 29 : : SEC  
 30 : : SBC #2 : RIGHT OF CENTER - ADJUST X  
 31 : : JMP BSW020 : GO SET X AND DRAW  
 32 : : LBS  
 33 : : ADC #2  
 34 : 6C35 BSW020  
 35 : 4C35 859A STA BUCKX  
 36 : 6C37 4C5760 JMP BSW050 : NOW DRAW IT  
 37 : : MOVE BUCK SHIP TO RIGHT WITH RIGHT TILT  
 38 :  
 39 : 6C3A BSW030  
 40 : 6C3A A902 LDA #2 : FORCE RIGHT TILT  
 41 : 6C3B 8580 STA LASTBS  
 42 : 6C3E A59A LDA BUCKX : ATTEMPT MOVE RT  
 43 : 6C40 18 CTC  
 44 : 6C41 6204 ADC #4  
 45 : 6C43 C9R1 CMP #177 : CHECK RIGHT LIMIT  
 46 : 6C45 B010 ^6C57 BCS BSW050 : TOO FAR RIGHT - DONT MOVE IT  
 47 : 6C47 4C5560 JMP BSW020 : SET NEW X AND DRAW  
 48 : : MOVE SHIP LEFT W/ LEFT TILT  
 49 :  
 50 : 6C4A BSW040  
 51 : 6C4A A900 LDA #0 : FORCE LEFT TILT  
 52 : 6C4B 8580 STA LASTBS  
 53 : 6C4E A59A LDA BUCKX : ATTEMPT MOVE LEFT  
 54 : 6C50 38 SEC  
 55 : 6C51 E904 SBC #4  
 56 : 6C53 C930 CMP #48

BUCK SHIP WORK

```

1 6C55 B0DE ^6C35 BCS BSW020 ; OK - SET X AND DRAW IT
2 ; POSITION BUCK SHIP BY Y
3
4 6C57 BSW050
5 6C57 A560 LDA JSDATA ; SEE IF ATTEMPTING UP OR DOWN
6 6C59 2904 AND #4
7 6C5B F012 ^6C6F BEQ BSW060 ; UP
8 6C5D A560 LDA JSDATA
9 6C5E 2908 AND #8
10 6C61 D033 ^6C96 BNE BSW100 ; NEITHER - XY SET - GO DRAW IT
11 ; MOVE BS DOWN
12 ;
13 ;
14 6C63 A59B LDA BUCKY ; ATTEMPT MOVE DOWN
15 6C65 18 CLC
16 6C66 6903 ADC #3
17 6C68 C981 CMP #129
18 6C6A B01E ^6C8A BCS BSW090 ; TOO LOW - SLOW DOWN SCROLL
19 6C6C 4C786C JMP BSW070 ; OK TO MOVE DOWN
20 ;
21 ; MOVE BS UP
22 ;
23 6C6F BSW060
24 6C6F A59B LDA BUCKY ; ATTEMPT MOVE UP
25 6C71 38 SEC
26 6C72 E903 SBC #3
27 6C74 C960 CMP #96
28 6C76 2005 ^6C7D BCC BSW080 ; TOO HIGH - SPEED UP SCROLL
29 6C78 BSW070
30 6C78 859B STA BUCKY ; SET NEW Y POSITION
31 6C7A 4C966C JMP BSW100 ; GO DRAW BS
32 ;
33 ; BUCK SHIP AT UPPER BOUNDARY - DO SPEED INCREASE
34 ;
35 6C7D BSW080
36 6C7D E699 INC SCRSPD ; INC SCROLL SPEED
37 6C7F A9C0 LDA #9C0
38 6C81 C599 CMP SCRSPD
39 6C83 B011 ^6C96 BCS BSW100 ; CHECK FF LIMIT
40 6C85 8599 STA SCRSPD
41 6C87 4C966C JMP BSW100 ; DRAW BS
42 ;
43 ; BUCK SHIP AT LOWER BOUNDARY - DO SPEED DECREASE
44 ;
45 6C8A BSW090
46 6C8A C699 DEC SCRSPD ; DOWN - DEC SCROLL SPEED
47 6C8C A599 LDA SCRSPD ; CHECK LOWER LIMIT
48 6C8E C5A7 CMP MINSPD
49 6C90 B004 ^6C96 BCS BSW100 ; OK
50 6C92 A5A7 LDA MINSPD ; SET AS LOWER LIMIT
51 6C94 8599 STA SCRSPD
52 ;
53 ; DRAW BUCK SHIP AND SHADOW
54 ;
55 6C96 BSW100
56 6C96 206A60 JSR CONVSP
57 6C99 4C6F72 JMP BSPAN ; PAINT BUCK SHIP, SHADOW AND EXIT

```

ROUTINE CHECKS FIRE BUTTON TO SEE IF TO FIRE SHOTS.  
 IF BUTTON IS HELD DOWN, A DELAY COUNTER REGULATES THE  
 SPEED AT WHICH SHOTS CAN BE FIRED. IF THE BUTTON IS  
 RELEASED, THE DELAY COUNTER IS RESET.

```

6C9C  A560      LDA    JSBATA
6C9E  2920      AND    #$20
6CA0  D007 ^6CA9  BNE    BSEI10 : BUTTON RELEASED
6CA2  A581      LDA    RSECNT : GET BUCK SHIP FIRE COUNT
6CA4  E008 ^6CAE  BEQ    BSEI20 : OK TO FIRE
6CA6  C481      DEC    RSECNT : DEC COUNTER AND EXIT
6CA8  60        RTS
6CA9  A900      LDA    #0      : RESET COUNTER WHEN BUTTON RELEASED
6CA9  A900      STA    RSECNT
6CA9  60        RTS

```

; CREATE BUCK SHOT AS TYPE 5, AT ((BUCKX AND \$F8)+8, BUCKY-3), USING PATHO.

```

6CAE  BSEI20
6CAE  A205      LDX    #5      : CREATE WITH TYPE 5.
6CB0  A000      LDY    #0      : SEARCH FROM TOP (BUCK SHOT GETS TOP 2)
6CB2  20E64C    JSR    CREA10 : CREATE OBJECT DESCRIPTOR
6CB5  D028 ^6CDF  BNE    BSFI30 : NO ROOM
6CB7  A59A      LDA    BUCKX : X=BUCKX+8
6CB9  19        CLC
6CBA  A908      ADC    #8
6CBB  990106    STA    OBJTBL+1,Y
6CBE  A59B      LDA    BUCKY : Y=BUCKY-3
6CB1  38        SEC
6CB2  E903      SBC    #3
6CB4  990206    STA    OBJTBL+2,Y
6CB7  A9C0      LDA    #$C0 : SPEED=12
6CC9  990506    STA    OBJTBL+5,Y
6CCC  A905      LDA    #5      : SIZE=5
6CCE  990306    STA    OBJTBL+3,Y
6CD1  A900      LDA    #0      : SET PATH POINTER
6CD3  200E6D    JSR    SETPTH
6CD6  A5AF      LDA    FIRELY : RESET FIRE DELAY COUNTER
6CD8  8581      STA    BSFCNT
6CDA  A907      LDA    #7
6CDE  20A863    JSR    SNDINI
6CDF  BSFI30
6CDE  60        RTS

```

ROUTINE CREATES A NEW OBJECT DESCRIPTOR INITIALIZING  
THE TYPE FROM X. RETURNS ACC=FF IF NO ROOM FOUND FOR  
OBJECT. OTHERWISE, ACC=0, Y-INDEX TO OBJECT DESCRIPTOR FOUND,  
TYPE BYTE IN OBJECT DESCRIPTOR IS SET FROM X GIVEN, AND ALL OTHER  
BYTES IN OBJECT DESCRIPTOR ARE SET=0. ENTRY POINT AT CREA10  
ALLOWS CREATING A PROCESS GIVEN THE SEARCH STARTING POINT IN Y.  
CALLS TO CREATE DO NOT LOOK AT TOP 2 OBJECT DESCRIPTOR LOCATIONS  
AS THEY ARE RESERVED FOR BUCK SHOTS.

```
0 6CE0      CREATE
1 6CE0  A010      LDY      #16    ; BEGIN SEARCH SKIPPING TOP 2 IN TABLE
2 6CE2      CREA5
3 6CE2  C000      CPY      #0    ; CHECK LIMIT
4 6CE4  F00D  ^6CF3  BEQ      CREA20 ; NO ROOM
5 6CE6  B90006  CREA10  LDA      OBJTBL,Y
6 6CE9  F00B  ^6CE6  BEQ      CREA30 ; FOUND SPACE
7 6CEB  98        TYA      : POINT TO NEXT SPOT
8 6CED  18        CLC
9 6CED  6908      ADC      #8
10 6CEF  A8        TAY
11 6CE0  4CE26C    JMP      CREA5
12
13 6CE3      CREA20
14 6CF3  A9FF      LDA      #$FF  ; NO ROOM
15 6CF5  40        RTS
16
17 6CF6      CREA30
18 6CF6  8A        TXA      : SET TYPE
19 6CF7  990006  STA      OBJTBL,Y
20 6CFA  A207      LDX      #7    ; SET OTHER 7 BYTES IN TABLE=0
21 6CFD  A900      LDA      #0
22 6CFE  C8        INY
23
24 6CFF      CREA40
25 6CFF  990006  STA      OBJTBL,Y
26 6D02  C8        INY
27 6D03  CA        DEX
28 6D04  D0F9  ^6CFF  BNE      CREA40
29 6D06  98        TYA      : Y -> TOP OF TABLE
30 6D07  38        SEC
31 6D08  E908      SBC      #8
32 6D0A  A8        TAY
33 6D0B  A900      LDA      #0    ; EVERYTHING SET
34 6D0D  60        RTS
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

1 : ROUTINE SETS PATH BYTES IN PTHPTR TO HEAD OF PATH # IN ACC  
 2 : Y=INDEX TO OBJECT DESCRIPTOR

3 : 0A-0A = Y SAVE

4 :

5 6D0E SETPTH

6 6D0E 840A STY 10 : SAVE Y

7 6D10 48 PHA : SAVE PATH #

8 6D11 28 TYA : GET POINTER/4 FOR PTHPTR INDEX

9 6D12 4A LSR A

10 6D13 4A LSR A

11 6D14 AA TAX : SET INDEX

12 6D15 48 PLA : GET BACK PATH#

13 6D16 0A ASL A : #2 FOR TABLE INDEX

14 6D17 A8 TAY : SET INDEX

15 6D18 B9C875 LDA PTHTR1,Y

16 6D19 9520 STA PTHPTR,X

17 6D1A B9C975 LDA PTHTR1+1,Y

18 6D20 9521 STA PTHTR+1,X

19 6D22 A40A LDY 10 : RESTORE Y

20 6D24 60 RTS

21 SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 48

22 SCROLL AND BACK DROP INITIALIZATION J:5200 .A65

23 : BACK DROP INITIALIZATION

24 : SETS UP MOUNTAIN RANGE AS CARDS 1-23, SCROLLING PATTERN  
 25 : CARDS' COLORS, MOUNTAIN CARDS' COLORS, AND BLANKS.

26 : CARD USAGE

27 : 00-00 = BLANK

28 : 01-18 = MOUNTAINS

29 : 19-F0 = BTMAP

30 : F1-FA = FUELSCORE

31 : FB-FB = SHIPS LEFT

32 : FC-F1 = UFO COUNT

33 : FD-FF = TIME GRAPH

34 : CONTROL REGISTERS INITIALIZED AS FOLLOWS:

35 6D25 BDINIT

36 6D25 A926 LDA #\$26

37 6D27 8D1109 STA COLOR0

38 6D2A A90F LDA #\$0F

39 6D2C 8D1209 STA COLOR1

40 6D2E A900 LDA #\$00

41 6D31 8D1309 STA COLOR2

42 6D34 8D1409 STA COLOR3

43 6D37 8D1509 STA COLOR4

44 6D3A 8500 STA 0

45 6D3C A920 LDA #\$20

46 6D3E 8501 STA 1

47 6D40 A000 LDY ##0

48 6D42 98 TYA

49 6D43 9100 STA [0],Y

50 6D44 98 DEY

6040	A000	LEY	#\$0
6042	98	TYA	
6043	BDIN20		
6043	9100	STA	[0],Y
6045	88	DEY	
6046	DOFB ^6043	BNE	BDIN20
6048	E601	TNC	
604A	A501	LDA	
604C	C940	CMP	#\$40
604E	DOE0 ^6040	BNE	BDIN10
6050	A205	LDX	#5
6052	BD766D	QAZL	LDA
6055	9D2920	STA	\$2029,X
6058	CA	DEX	
6059	10E7 ^6052	BPL	QAZL
605B	A205	LDX	#5
605D	BD7C6D	QAZK	LDA
6060	9D3D20	STA	\$203D,X
6063	CA	DEX	
6064	10E7 ^605D	BPL	QAZK
6066	A5AC	LDA	SYSTAT
6068	2902	AND	#%00000010
606A	DO04 ^6070	BNE	SKIPMT
606C	20B26B	ISR	MTNW90
606E	20B26B	ISR	MTNW90
6070	20B26B	ISR	MTNW90
6072	20B26B	ISR	MTNW90
6074	20B26B	ISR	MTNW90
6076	20B26B	ISR	MTNW90
6078	20B26B	ISR	MTNW90
6080	20B26B	ISR	MTNW90
6082	20B26B	ISR	MTNW90
6084	20B26B	ISR	MTNW90
6086	20B26B	ISR	MTNW90
6088	20B26B	ISR	MTNW90
6090	20B26B	ISR	MTNW90
6092	20B26B	ISR	MTNW90
6094	20B26B	ISR	MTNW90
6096	20B26B	ISR	MTNW90
6098	20B26B	ISR	MTNW90
6100	20B26B	ISR	MTNW90
6102	20B26B	ISR	MTNW90
6104	20B26B	ISR	MTNW90
6106	20B26B	ISR	MTNW90
6108	20B26B	ISR	MTNW90
6110	20B26B	ISR	MTNW90
6112	20B26B	ISR	MTNW90
6114	20B26B	ISR	MTNW90
6116	20B26B	ISR	MTNW90
6118	20B26B	ISR	MTNW90
6120	20B26B	ISR	MTNW90
6122	20B26B	ISR	MTNW90
6124	20B26B	ISR	MTNW90
6126	20B26B	ISR	MTNW90
6128	20B26B	ISR	MTNW90
6130	20B26B	ISR	MTNW90
6132	20B26B	ISR	MTNW90
6134	20B26B	ISR	MTNW90
6136	20B26B	ISR	MTNW90
6138	20B26B	ISR	MTNW90
6140	20B26B	ISR	MTNW90
6142	20B26B	ISR	MTNW90
6144	20B26B	ISR	MTNW90
6146	20B26B	ISR	MTNW90
6148	20B26B	ISR	MTNW90
6150	20B26B	ISR	MTNW90
6152	20B26B	ISR	MTNW90
6154	20B26B	ISR	MTNW90
6156	20B26B	ISR	MTNW90
6158	20B26B	ISR	MTNW90
6160	20B26B	ISR	MTNW90
6162	20B26B	ISR	MTNW90
6164	20B26B	ISR	MTNW90
6166	20B26B	ISR	MTNW90
6168	20B26B	ISR	MTNW90
6170	20B26B	ISR	MTNW90
6172	20B26B	ISR	MTNW90
6174	20B26B	ISR	MTNW90
6176	20B26B	ISR	MTNW90
6178	20B26B	ISR	MTNW90
6180	20B26B	ISR	MTNW90
6182	20B26B	ISR	MTNW90
6184	20B26B	ISR	MTNW90
6186	20B26B	ISR	MTNW90
6188	20B26B	ISR	MTNW90
6190	20B26B	ISR	MTNW90
6192	20B26B	ISR	MTNW90
6194	20B26B	ISR	MTNW90
6196	20B26B	ISR	MTNW90
6198	20B26B	ISR	MTNW90
6200	20B26B	ISR	MTNW90
6202	20B26B	ISR	MTNW90
6204	20B26B	ISR	MTNW90
6206	20B26B	ISR	MTNW90
6208	20B26B	ISR	MTNW90
6210	20B26B	ISR	MTNW90
6212	20B26B	ISR	MTNW90
6214	20B26B	ISR	MTNW90
6216	20B26B	ISR	MTNW90
6218	20B26B	ISR	MTNW90
6220	20B26B	ISR	MTNW90
6222	20B26B	ISR	MTNW90
6224	20B26B	ISR	MTNW90
6226	20B26B	ISR	MTNW90
6228	20B26B	ISR	MTNW90
6230	20B26B	ISR	MTNW90
6232	20B26B	ISR	MTNW90
6234	20B26B	ISR	MTNW90
6236	20B26B	ISR	MTNW90
6238	20B26B	ISR	MTNW90
6240	20B26B	ISR	MTNW90
6242	20B26B	ISR	MTNW90
6244	20B26B	ISR	MTNW90
6246	20B26B	ISR	MTNW90
6248	20B26B	ISR	MTNW90
6250	20B26B	ISR	MTNW90
6252	20B26B	ISR	MTNW90
6254	20B26B	ISR	MTNW90
6256	20B26B	ISR	MTNW90
6258	20B26B	ISR	MTNW90
6260	20B26B	ISR	MTNW90
6262	20B26B	ISR	MTNW90
6264	20B26B	ISR	MTNW90
6266	20B26B	ISR	MTNW90
6268	20B26B	ISR	MTNW90
6270	20B26B	ISR	MTNW90
6272	20B26B	ISR	MTNW90
6274	20B26B	ISR	MTNW90
6276	20B26B	ISR	MTNW90
6278	20B26B	ISR	MTNW90
6280	20B26B	ISR	MTNW90
6282	20B26B	ISR	MTNW90
6284	20B26B	ISR	MTNW90
6286	20B26B	ISR	MTNW90
6288	20B26B	ISR	MTNW90
6290	20B26B	ISR	MTNW90
6292	20B26B	ISR	MTNW90
6294	20B26B	ISR	MTNW90
6296	20B26B	ISR	MTNW90
6298	20B26B	ISR	MTNW90
6300	20B26B	ISR	MTNW90
6302	20B26B	ISR	MTNW90
6304	20B26B	ISR	MTNW90
6306	20B26B	ISR	MTNW90
6308	20B26B	ISR	MTNW90
6310	20B26B	ISR	MTNW90
6312	20B26B	ISR	MTNW90
6314	20B26B	ISR	MTNW90
6316	20B26B	ISR	MTNW90
6318	20B26B	ISR	MTNW90
6320	20B26B	ISR	MTNW90
6322	20B26B	ISR	MTNW90
6324	20B26B	ISR	MTNW90
6326	20B26B	ISR	MTNW90
6328	20B26B	ISR	MTNW90
6330	20B26B	ISR	MTNW90
6332	20B26B	ISR	MTNW90
6334	20B26B	ISR	MTNW90
6336	20B26B	ISR	MTNW90
6338	20B26B	ISR	MTNW90
6340	20B26B	ISR	MTNW90
6342	20B26B	ISR	MTNW90
6344	20B26B	ISR	MTNW90
6346	20B26B	ISR	MTNW90
6348	20B26B	ISR	MTNW90
6350	20B26B	ISR	MTNW90
6352	20B26B	ISR	MTNW90
6354	20B26B	ISR	MTNW90
6356	20B26B	ISR	MTNW90
6358	20B26B	ISR	MTNW90
6360	20B26B	ISR	MTNW90
6362	20B26B	ISR	MTNW90
6364	20B26B	ISR	MTNW90
6366	20B26B	ISR	MTNW90
6368	20B26B	ISR	MTNW90
6370	20B26B	ISR	MTNW90
6372	20B26B	ISR	MTNW90
6374	20B26B	ISR	MTNW90
6376	20B26B	ISR	MTNW90
6378	20B26B	ISR	MTNW90
6380	20B26B	ISR	MTNW90
6382	20B26B	ISR	MTNW90
6384	20B26B	ISR	MTNW90
6386	20B26B	ISR	MTNW90
6388	20B26B	ISR	MTNW90
6390	20B26B	ISR	MTNW90
6392	20B26B	ISR	MTNW90
6394	20B26B	ISR	MTNW90
6396	20B26B	ISR	MTNW90
6398	20B26B	ISR	MTNW90
6400	20B26B	ISR	MTNW90
6402	20B26B	ISR	MTNW90
6404	20B26B	ISR	MTNW90
6406	20B26B	ISR	MTNW90
6408	20B26B	ISR	MTNW90
6410	20B26B	ISR	MTNW90
6412	20B26B	ISR	MTNW90
6414	20B26B	ISR	MTNW90
6416	20B26B	ISR	MTNW90
6418	20B26B	ISR	MTNW90
6420	20B26B	ISR	MTNW90

```

1 6D6F 60
2 6D70 A940 RTS
3 6D72 800E04 SKIPMT LDA #%1000000
4 6D75 60 STA NMEN
5
6 6D76 514B4E504C SCDAT DB $1,75,79,80,76,83
7 6D7C C002CCCE00 T1DAT DB 77,0R $C0,82,0R $C0,76,0R $C0,78,0R $C0,0,83,0R $C0
8
9 6D82 A2C0 SETDLI LDIX #%11000000
10 6D84 A900 LDA #$0
11 6D86 8DB600 STA SCRPLTR
12 6D88 A940 LDA #%01000000
13 6D8B 8D0E04 STA NMEN
14 6D8E A00B04 SDLOP LDA VCOUNT
15 6D91 C903 CMP #3
16 6D93 D0E2 ^A18E BNE SDLOP
17 6D95 8E0E04 STX NMEN
18 6D98 60 RTS
19
20 6D99 A5AE SETCOL LDA CLEVYL
21 6D9B 4A LSR A
22 6D9C 4A LSR A
23 6D9D AA TAX
24 6D9E B00C60 LUDCOL LDA COLRSL,X
25 6DA1 8504 STA 4
26 6DA3 BDE16D LDA COLRSH,X
27 6DA6 8505 STA 5
28 6DA8 A235 LDA #LOW SCROL0
29 6DAA 8500 STA 0
30 6DAC A27B LDA #HIGH SCROL0
31 6DAE 8501 STA 1
32 6DB0 A200 LDA #LOW RELSCR
33 6DB2 8502 STA 2
34 6DB4 A910 LDA #HIGH RELSCR
35 6DB6 8503 STA 3
36 6DB8 A000 XLOOP LDY #$0
37 6DBA B100 LDY C01,Y
38 6DBC A8 TAY
39 6DBD B104 LDA C41,Y
40 6DBF A000 LDY #$0
41 6DC1 9102 STA C21,Y
42 6DC3 E600 INC 0
43 6DC5 D002 ^6DC9 BNE GL79
44 6DC7 E601 INC 1
45 6DC9 E602 GL79 INC 2
46 6DCB D002 ^6DCE BNE GL710
47 6DCD E603 INC 3
48 6DCE A501 GL710 LDA 1
49 6DD1 C97D CMP #HIGH SCREND
50 6DD3 D0E3 ^6DB8 BNE XLOOP
51 6DD5 A500 LDA 0
52 6DD7 C915 CMP #LOW SCREND
53 6DD9 D0D0 ^6DB8 BNE XLOOP
54 6DDB 60 RTS
55 6DDC F0E2F4E6E8 COLRSI DB LOW (CLEV1-6),LOW (CLEV2-6),LOW (CLEV3-6),LOW (CLEV4-6),LOW (CLEV5-6)
56 6DE1 6D6D6D6D6D COLRSH DB HIGH (CLEV1-6),HIGH (CLEV2-6),HIGH (CLEV3-6),HIGH (CLEV4-6),HIGH (CLEV5-6)
57 6DE4 9E90 CLEV1 DB $98,$90
58 6DE8 1A74 CLEV2 DB $1A,$74
59 6DEA 1DD4 CLEV3 DB $1D,$D4
60 6DEC 0604 CLEV4 DB $06,$04

```

60EE D8D4 CLEV5 DB \$D8,\$D4

1 ; EACH OBJECT IS HANDLED SEPARATELY ACCORDING TO TYPE  
2 ; THE PC TYPE IS USED TO INDEX INTO A TABLE TO LOCATE THE  
3 ; PROPER DRIVER. PC TYPES START AT 1. IF Z=9, THEN OBJECT  
4 ; IS NOT TO BE DISPLAYED AND A FLAG IS SET.  
5 ;  
6 ;  
7 ; IN ALL CASES, X0=X, Y0=Y  
8 ;  
9 ; ENTER:  
10 ;  
11 ; Y=INDEX INTO OBJTBL FOR OBJECT DESCRIPTOR  
12 ;  
13 ; EXIT:  
14 ;  
15 ; 00-01 = USED  
16 ;  
17 ; A=0 IS DISPLAYABLE  
18 ; A=FF IS NOT DISPLAYABLE  
19 ;  
20 6DF0 890306 PCXYAC LDA DBJTBL+3,Y ; SEE IF INDISPLAYABLE  
21 6DF0 890306 LDA DBJTBL+3,Y ; SEE IF INDISPLAYABLE  
22 6DF3 C909 CMP #9  
23 6DF5 9003 ^6DFA BCC PCXY10  
24 6DF7 PCXY05  
25 6DF7 A9FF LDA #\$FF ; SET NON-DISPLAYABLE FLAG  
26 6DF9 60 RTS  
27 6DFA PCXY10  
28 6DFA 890206 LDA DBJTBL+2,Y  
29 6DFA C900 CMP #144  
30 6DFF B0E6 ^6DFA BCS PCXY05 ; BELOW SCREEN - FORGET IT  
31 6E01 B90006 LDA DBJTBL+0,Y  
32 6E04 AA TAX  
33 6E05 B00079 LDA MCODEL-1,X  
34 6E08 8500 STA 0  
35 6E0A B00279 LDA MCODEH-1,X  
36 6E0D 8501 STA 1  
37 6E0F B90306 LDA DBJTBL+3,Y  
38 6E12 A8 TAY  
39 6E13 B100 LDA [\$001],Y  
40 6E15 856B STA ADDR  
41 6E17 98 TYA  
42 6E18 18 CLC  
43 6E19 690A ADC #10  
44 6E1B A8 TAY  
45 6E1C B100 LDA [\$001],Y  
46 6E1E 856C STA ADDR+1  
47 6E20 A40A LDY 10  
48 6E22 B90006 LDA DBJTBL+0,Y  
49 6E25 AA TAX  
50 6E26 B0827A LDA MCODEL-1,X  
51 6E27 8500 STA 0  
52 6E2B B0887A LDA MCODEH-1,X  
53 6E2E 8501 STA 1  
54 6E30 B90306 LDA DBJTBL+3,Y  
55 6E33 A8 TAY  
56 6E34 B100 LDA [\$001],Y  
57 6E36 856D STA ADDR  
58 6E38 98 TYA  
59 6E39 18 CLC

```
1 6E3A 690A ADC #10
2 6E3C A8 TAY
3 6E3D B100 LDA [$0033,Y]
4 6E3E 856E STA ADDRM#1
5 6E41 A40A LDY 10
6 6E43 B90104 LDA OBJTBL+1,Y
7 6E46 851D STA $1D
8 6E48 B90206 LDA OBJTBL+2,Y
9 6E4B 851E STA $1E
10 6E4D A900 LDA #$0
11 6E4F 60 RTS
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
```

1 : FOR NORMAL SCALLED OBJECTS, PC TYPE IS USED TO INDEX  
2 : INTO A TABLE POINTING TO THE BASE OF A 36 BYTE TABLE  
3 : CONTAINING ALL THE PC NUMBERS FOR ALL 4 DIFFERENT Y/TL VALUES  
4 : AND ALL 9 POSSIBLE SIZES OF PICTURES. Z IS THEN USED TO GET  
5 : THE ONE OF FOUR PICTURES PER SIZE. THIS CREATES THE  
6 : SMOOTH SCROLLING.

7 :  
8 : ENTER:  
9 :  
10 :  
11 : Y=INDEX INTO OBJECT DESCRIPTOR  
12 : X=OBJECT TYPE INDEX (PC TYPE\*2)

13 6E50 PC\$CAL  
14 6E50 98 TYA  
15 6E51 48 PHA  
16 6E52 B90006 LDA OBJTBL+0,Y  
17 6E55 AA TAX  
18 6E56 B0D079 LDA DCODE1-1,X  
19 6E59 8500 STA 0  
20 6E5B B0D979 LDA DCODEH-1,X  
21 6E5E 8501 STA 1  
22 6E60 B90306 LDA OBJTBL+3,Y  
23 6E63 A8 TAY  
24 6E64 B100 LDA F01,Y  
25 6E66 856B STA ADDR  
26 6E68 98 TYA  
27 6E69 18 CLC  
28 6E6A 690A ADC #\$A  
29 6E6C A8 TAY  
30 6EAD B100 LDA F01,Y  
31 6E6F 856C STA ADDR+1  
32 6E71 A009 LDY #\$9  
33 6E73 B16B LDA FADDR1,Y  
34 6E75 850B STA \$B  
35 6E77 C8 INY  
36 6E78 B16B LDA FADDR1,Y  
37 6E7A 850C STA \$C  
38 6E7C 68 PLA  
39 6E7D A8 TAY  
40 6E7E 60 RTS  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

FTBL FOR PIX SIZE

1 ; ROUTINE MOVES ACTIVE OBJECTS ABOUT DISPLAY AREA  
2 ; OBJECT DATA BASE CONTAINS 8 BYTES PER OBJECT  
3 ; ORGANIZED AS FOLLOWS:  
4 ;  
5 ; 0 = PICTURE CODE (PC)  
6 ; 1 = X LOCATION (0-255)  
7 ; 2 = Y LOCATION (0-255)  
8 ; 3 = Z (SIZE 0-2)  
9 ; 4 = COUNT  
10 ; 5 = SPEED  
11 ; 6 = REPEAT COUNT  
12 ; 7 = STATUS  
13 ; 7 = MOVED ALREADY  
14 ; 6 =  
15 ; 5 =  
16 ; 4 = MOTHER ZORBA ATTACKING/RUCK SHOT ON REBOUND FROM ZORBA  
17 ; 3 = LEFT POLE FLAG  
18 ; 2 =  
19 ; 1 =  
20 ; 0 =  
21 ;  
22 ;  
23 ; MOVES FURTHEST OBJECTS FIRST, FOLLOWED BY CLOSER OBJECTS.  
24 ;  
25 ; PC REPRESENTS THE OBJECT TYPE (EG. POLE, SAUCER, ETC.) AND  
26 ; IS TRANSLATED INTO THE PROPER PICTURE TO BE USED BY PAINT  
27 ; AND ERASE. (X, Y, Z) IS TRANSLATED TO THE DISPLAY EQUIVALENT  
28 ; FORM OF (XD, YD) AND THEN PASSED ON TO PAINT/ERASE.  
29 ;  
30 ;  
31 ; ENTER:  
32 ;  
33 ; EXIT:  
34 ;  
35 ;  
36 ; 00-07 = USED BY PAINT/ERASE  
37 ; 08-08 = TEMP FOR MOVE  
38 ; 09-09 = INCOMPLETE FOUND FLAG  
39 ; 0A-0A = OBJECT TABLE INDEX TO USE  
40 ; 10-1B = USED BY PAINT/ERASE  
41 ; 1C-1C = PC NUMBER TO DISPLAY  
42 ; 1D-1D = XD  
43 ; 1E-1E = YD  
44 ; 1F-1F = FG COLOR  
45 ;  
46 6E7F MOVE  
47 6E7F A000 LDIY #0 ; SET UP INDEXES  
48 6E81 A220 LDIX #32  
49 6E83 A9FF LDA #\$FF  
50 6E85 8509 STA 8  
51 6E87 A900 LDA #0  
52 6E89 8509 STA 9  
53 6E8B MOVE20  
54 6E8B R80006 LDA ORJTRL,Y ; GET PIC CODE TYPE  
55 6E8E F017 ^6EA7 BEQ MOVE30 ; IF ZERO THEN GET NEXT ENTRY  
56 6E90 B90706 LDA OBJTBL+7,Y ; CHECK HI BIT OF STATUS REGISTER  
57 6E93 3012 ^6E87 BMI MOVE30 ; DONE ALREADY  
58 6E95 A9FF LDA #\$FF ; FOUND ONE INCOMPLETE  
59 6E97 8509 STA 9 ; SET FLAG

6E95 A9FF  
6E97 8509

LDA #FF  
STA 9

FOUND ONE INCOMPLETE  
; SET FLAG

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 55  
ROUTINE FOR MOVING OBJECTS ABOUT DISPLAY J:5200 .A65

6E99 A508 LDA 8 ; GET LAST SMALLEST Y  
6E9B D90206 CMP DBJTBL+2,Y  
6E9E 5007 ^6EA7 BCD MOVE30 ; THIS ONE IS CLOSER THAN LAST  
6EA0 B90206 LDA DBJTBL+2,Y ; THIS IS FURTHEST SO FAR  
6EA3 E508 STA 8  
6EA5 840A STY 10 ; USE THIS ONE  
6EA7 MOVE30  
6EA7 98 TYA ; CHECK NEXT  
6EAB 18 CLC  
6EA9 A908 ADC #8  
6EAB A8 TAY  
6EAC CA DEX  
6EAD B00C ^6E8B BNE MOVE20 ; DO IT 16 TIMES  
6EAF A509 LDA 9 ; SEE IF FOUND ANY INCOMPLETE  
6EB1 D020 ^6ED3 BNE MOVE40 ; YES - PROCESS THIS ONE  
6ER3 A000 LDY #0 ; CLEAR ALL MOVED BITS FOR NEXT TIME  
6EB5 A220 LDX #32  
6ER7 MOVE35  
6ER7 A97F LDA #\$7F  
6ER9 390706 AND DBJTBL+7,Y  
6EB5 990706 STA DBJTBL+7,Y  
6ERF 98 TYA  
6EC0 18 CLC  
6EC1 6208 ADC #8  
6EC3 A8 TAY  
6EC4 CA DEX  
6EC5 D0F0 ^6EB7 BNE MOVE35  
6EC7 A57B LDA LVLELG  
6EC9 E007 ^6ED2 BEQ NOGAIN  
6ECB A200 LDA #\$0  
6ECTI 857B STA LVLELG  
6ECE 400C64 JMP LVLOVR  
6ED2 60 NOGAIN RTS  
6ED3 MOVE40  
6ED3 A40A LDY 10 ; GET INDEX INTO DBJTBL  
6ED5 B90706 LDA DBJTBL+7,Y ; SET AS OBJECT COMPLETED  
6ED8 0980 ORA #\$80  
6EDA 990706 STA DBJTBL+7,Y  
6EDD 20F06D JSR PCXYAC ; SET UP X0, Y0, PC AND COLOR  
6EE0 D003 ^6EES BNE MOVE50 ; NOT TO BE ERASED  
6EE2 201868 JSR ERASE ; ERASE CURRENT  
6EES MOVE50  
6EES A40A LDY 10 ; RESET INDEX  
6EE7 20036F JSR MOVPOS ; MOVE OBJECT POSITION  
6EEA D00D ^6EF9 BNE MOVE70  
6EEC A40A LDY 10 ; RESET INDEX  
6EEE 20F06D JSR PCXYAC ; SET UP X0, Y0, PC AND COLOR  
6EF1 D003 ^6EF6 BNE MOVE40 ; NOT TO BE DRAWN  
6EF3 205A67 JSR PAINT ; DRAW IT  
6EF6 MOVE60  
6EF6 4C7F6E JMP MOVE ; GO DO NEXT ONE  
6EF9 MOVE70  
6EE9 A40A LDY 10 ; RESET INDEX  
6EEB A200 LDA #0 ; SET PC=0 TO CANCEL OBJECT  
6EFL 920004 STA DBJTBL,Y  
6F00 4C7F6E JMP MOVE ; NEXT...

1 ; MOVE OBJECT SPEED/16 PIXELS IN THE DIRECTION OF THE  
 2 ; DIRECTION BYTE. THE AMOUNT TO MOVE IN ANY PASS IS CALCULATED  
 3 ; AS FOLLOWS:

4 ;  
 5 ; DISTANCE=((OLD-COUNT+SPEED)/16)+((SCRSPD\*(9-Z))/256)-(OLD-COUNT/16)  
 6 ;

7 ; ENTER:

8 ;  
 9 ; Y INDEX SET TO OBJECT DESCRIPTOR

10 ;  
 11 ; EXIT:

12 ;  
 13 ; 00-04 = USED

14 ; 03-03 = DIRECTION BYTE SAVE

15 ; 04-04 = X INDEX SAVE (Y/4)

16 ;  
 17 ;  
 18 6E03 MOVPOS  
 19 6E03 98 TYA ; SET UP X FOR PTHPTR  
 20 6E04 4A LSR A ; =OBJTBL INDEX/4  
 21 6E05 4A LSR A  
 22 6E06 AA TAX ; GET TO X  
 23 6E07 8604 SIX 4 ; SAVE INDEX  
 24 6E09 MOVPO3  
 25 6F09 B90606 LDA OBJTBL+6,Y ; GET RPTCNT  
 26 6F0C D006 ^6F14 BNE MOVPOS ; OK TO MOVE  
 27 6E0E 200870 JSR NXTRPT ; SET POINTERS  
 28 6F11 F0E6 ^6F09 BEQ MOVPO3 ; NOW MOVE IT  
 29 6F13 60 RTS ; GONE  
 30 ;  
 31 ; GET DISTANCE TO MOVE  
 32 ;  
 33 6F14 MOVPOS  
 34 6F14 B90006 LDA OBJTBL,Y ; IF BUCK SHOT, SCRSPD OFFSET=0  
 35 6F17 C905 CMP #5  
 36 6F19 D007 ^6F22 BNE MOVPO6  
 37 6F1B A900 LDA #0 ; SET SCRSPD OFFSET=0  
 38 6F1D 8502 STA 2  
 39 6F1F 4C316F JMP MOVPO7  
 40 6F22 MOVPO6  
 41 6F22 A909 LDA #2 ; CALCULATE SCRSPD\*(9-Z)/256 FOR Y ADJUST  
 42 6F24 38 SEC  
 43 6F25 F90306 SBC OBJTBL+3,Y  
 44 6F28 A699 LDX SCRSPD  
 45 6F2A 20C273 JSR MULT  
 46 6F2D 8502 STA 2 ; A=1/256  
 47 6F2F 0602 ASL 2  
 48 6F31 MOVPO7  
 49 6F31 A604 LDX 4 ; RESTORE X  
 50 6F33 B90406 LDA OBJTBL+4,Y ; GET OLD COUNT  
 51 6F34 8501 STA 1 ; SAVE OLD COUNT  
 52 6F38 4A LSR A ; GET OLD COUNT/16  
 53 6F39 4B LSR A  
 54 6F3A 4A LSR A  
 55 6F3B 4A LSR A  
 56 6F3C 8500 STA 0 ; SET OLD COUNT/1NT  
 57 6F3E A501 LDA 1 ; GET BACK OLD COUNT  
 58 6F40 18 CLC ; GET (OLD COUNT+SPEED)/16  
 59 6F41 790506 ADC OBJTBL+5,Y

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 57  
MOVE POSITION OF OBJECT J:5200 .A65

```

6F44 990406 STA OBJTBL+4, Y ; UPDATE OLD COUNT
6F47 6A ROR
6F48 4A LSR
6F49 4A LSR
6F4A 4A LSR
6F4B 18 CLC
6F4C A502 ADC
6F4E 38 SEC
6F4F E500 SBC 0 ; ALWAYS < 256
6F51 MOVE10
6F51 8500 STA 0 ; SAVE
6F53 A604 LDX 4 ; GET INDEX INTO PTHPTR
6F55 A120 LDA [PTHPTR,X1] ; GET DIRECTION BYTE
6F57 8503 STA
6F59 A500 LDA
6F5B F02B ^6F88 BEQ MOVP40 ; NOWHERE TO MOVE
6F5D 38 SEC
6F5E E20406 SBC OBJTBL+6, Y ; -RPTCNT
6F61 B00F ^6F72 BCS MOVP20 ; DISTANCE=RPTCNT
; DISTANCE TO MOVE LESS THAN RPTCNT VALUE. SIMPLY REPEAT.
; DTR MOVE (DISTANCE) TIMES AND REDUCE RPTCNT BY DISTANCE.

```

```

6F63 49FF EOR #$FF ; NEW RPTCNT=RPTCNT-DISTANCE
6F65 6901 ADC #1 ; C=0
6F67 990606 STA OBJTBL+6, Y ; UPDATE REPEAT COUNT
6F6A 20D46F JSR MVXY ; MOVE IN XY BY DISTANCE IN 0
6F6D D059 ^6FC8 BNE MOV100 ; GONE
6F6F 4C886F JMP MOVP40 ; DO X. MTNSPD OFFSET AND EXIT
; DISTANCE TO MOVE GREATER THAN (OR EQUAL TO) RPTCNT. FINISH
; OUT REMAINING RPTCNT MOVES AND GET NEW REPEAT COUNT/DIR. BYTE
; AND FINISH OUT DISTANCE COUNT

```

```

6F72 MOVP20
6F72 8502 STA 2 ; SAVE DISTANCE REMAINING
6F74 B20406 LDA OBJTBL+6, Y ; GET CURRENT RPTCNT
6F77 8500 STA 0 ; SET AS DISTANCE TO MOVE HERE
6F79 20D46F JSR MVXY ; FINISH OUT THIS RPTCNT
6F7C D04A ^6FC8 BNE MOV100 ; GONE
6F7E 200870 JSR NXTRPT ; GET NEXT REPEAT MOVE & REPEAT COUNT
6F81 D045 ^6FC8 BNE MOV100 ; GONE
6F83 A502 LDA 2 ; GET REMAINING FOR THIS MOVE
6F85 4C516F JMP MOVP10
; DO X ADJUSTMENT BY MTNSPD.
; ADJUSTMENT ONLY FOR HOPPERS, POLES AND SAUCERS

```

```

6F88 B90006 LDA OBJTBL, Y ; SEE WHAT THIS IS
6F8B C904 CMP #4
6F8D B036 ^6FC5 BCS MOVP20 ; NO MTNSPD ADJUSTMENT
6F8E BE0306 LDX OBJTBL+3, Y ; GET SIZE TO X
6F92 E009 CPX #9 ; NO X ADJUSTMENT FOR SIZE 9
6F94 E02E ^6FC5 BEQ MOVP20
6F96 BD0B6F LDA XSHIFT, X ; GET SHIFT COUNT FROM TABLE
6F99 8500 STA 0 ; DIVIDE MTNSPD BY 8*LOJ
6F9B A596 LDA MTNSPD ; SAVE SIGN OF MTNSPD
6F9D 08 PHP ; FORCE POSITIVE #
6F9E 1005 ^6FA5 BPL MOVP50 ; FORCE POSITIVE #

```

```

1 6FA0 49FF EOR #$FF
2 6FA2 18 CLC
3 6FA3 4901 ADC #1
4 6FA5 MOVEP50
5 6FA5 4A LSR A ; DIVIDE BY 2
6 6FA6 C600 DEC 0
7 6FA8 D0FB ^6FA5 BNE MOVEP50
8 6FAA MOVEP60
9 6FAA 28 PLP ; GET SIGN OF MTNSPD
10 6FAB 1005 ^6FB2 BPL MOVEP70
11 6FAE 49FF EOR #$FF ; MAKE NEG AGAIN
12 6FAF 18 CLC
13 6FB0 4901 ADC #1
14 6FB2 MOVEP70
15 6FB2 08 PHP ; SAVE SIGN OF MTNSPD
16 6FB3 18 CLC ; ADD TO X POSITION
17 6FB4 790106 ADC ORJtbl+1,Y
18 6FB7 990106 STA ORJtbl+1,Y
19 6F8A B006 ^6FC2 BCS MOVEP80 ; CHECK BOUNDARIES
20 6F8C 28 PLP ; GET BACK SIGN
21 6F8D 3009 ^6FC8 BMI MOV100 ; OUT OF AREA
22 6F8F 40054F NMP MOVEP90 ; STILL OK
23 6FC2 MOVEP80
24 6FC2 28 PLP ; GET BACK MTNSPD SIGN
25 6FC3 1003 ^6FC8 BPL MOVE100 ; OUT OF AREA
26 6FC5 MOVEP90
27 6FC5 A900 LDA #0 ; SET STILL IN BOUNDS
28 6FC7 60 RTS
29 6FC8 MOVE100
30 6FC8 A9FF LDA #$FF ; OUT OF DISPLAY - TO BE CANCELED
31 6FC9 60 RTS
32 6FCB XSHIFT DB 2,3,3,4,4,4,5,5,5
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

1 ; MOVE XY LOCATIONS IN UNITS GIVEN DIR BYTE IN D

2 ;

3 6FD4 MVXY

4 6FD4 20E36F JSR MVXY30 ; MOVE X

5 6FD7 D009 ^6FF2 BNE MVXY20 ; GONE

6 6FD9 C8 INY ; SET UP TO MOVE Y

7 6FDA 20E36F JSR MVXY30

8 6FDD D019 ^6FF8 BNE MVXY40 ; GONE

9 6EDE 88 DEY ; RESET Y

10 6EE0 MVXY10

11 6FE0 A900 LDA #0 ; STILL OK

12 6FE2 MVXY20

13 6FE2 60 RTS

14 6FE3 MVXY30

15 6FE3 2403 ROL 3 ; SEE IF TO MOVE THIS ONE

16 6FE5 90E9 ^6FE0 BCC MVXY10 ; NO - STILL OK

17 6EE7 A910 LDA #14 ; POS/NEG?

18 6FE9 2503 AND 3

19 6FEB F00E ^6FFB BEQ MVXY40 ; NEG

20 6FED B90106 LDA DRJtbl+1,Y ; POSITIVE MOVE

21 6FE0 18 CLC ; ADD IN DISTANCE

22 6FF1 4500 ADC 0

23 6FE3 990106 STA DRJtbl+1,Y

24 6FF6 90E8 ^6FE0 BCC MVXY10 ; OK

25 6FF8 MVXY40

26 6FF8 A9FF LDA #\$FF ; GONE

27 6FFA 60 RTS

28 6FEB MVXY50

29 6FEB B90106 LDA DRJtbl+1,Y ; SUBTRACT OUT DISTANCE

30 6FFF F500 SBC 0 ; C=1

31 7000 990106 STA DRJtbl+1,Y

32 7003 B0B8 ^6FE0 BCS MVXY10 ; OK

33 7005 4CFF4F JMP MVXY40 ; GONE

1 ; GETS NEXT REPEAT COUNT, HANDLING ANY SPECIAL CONFIGURATION BYTES  
 2  
 3 ; MOVE ROUTINE MOVES AN OBJECT ACCORDING TO THE DIRECTION  
 4 ; AND SPEED. SPEED REPRESENTS THE NUMBER OF SIXTEENTHS OF  
 5 ; A PIXEL THAT THE OBJECT IS TO MOVE IN THE DIRECTION SPECIFIED  
 6 ; BY THE DIRECTION BYTE IN THE PATH TABLE. THE PATH TABLE IS POINTED TO  
 7 ; BY A 32 BYTE TABLE (DIRTBP) IN 0-PAGE (2 BYTES/OBJECT) USED FOR  
 8 ; INDEXING INTO THE PATH TABLE AND THE INDEX VALUE NORMALLY  
 9 ; POINTS TO THE DIRECTION BYTE. THE ORDER OF THE PATH TABLE  
 10 ; IS SIMPLY A REPEAT COUNT BYTE (0-127) FOLLOWED BY THE DIRECTION  
 11 ; OF TRAVEL. WHEN A NEW DIRECTION BYTE IS READ, THE SPEED AND Z  
 12 ; VALUES ARE UPDATED. AFTER THAT, ONLY THE X AND Y VALUES ARE  
 13 ; CHANGED. THE DIRECTION BYTE DEFINES THE DIRECTION OF TRAVEL  
 14 ; AS FOLLOWS:  
 15  
 16 ;  
 17 ; BIT 0 = +/-SPEED 1=+, 0=-  
 18 ; BIT 1 = +/-Z (SIZE) 1=+, 0=-  
 19 ; BIT 2 = +/-Y 1=+, 0=-  
 20 ; BIT 3 = +/-X 1=+, 0=-  
 21 ; BIT 4 = SPEED CHANGE 1=CHANGE, 0=NO CHANGE  
 22 ; BIT 5 = Z CHANGE 1=CHANGE, 0=NO CHANGE  
 23 ; BIT 6 = Y CHANGE 1=CHANGE, 0=NO CHANGE  
 24 ; BIT 7 = X CHANGE 1=CHANGE, 0=NO CHANGE  
 25 ;  
 26 ; IF COUNT BIT 7=1, THEN IT IS TO BE INTERPRETED AS A SPECIAL  
 27 ; CONFIGURATION BYTE FOLLOWED BY ANY ARGUMENTS REQUIRED.  
 28 ; OF THE CONFIGURATION COMMAND. THE CONFIGURATION COMMANDS  
 29 ; ARE AS FOLLOWS:  
 30 ;  
 31 ; 80 = JUMP TO adr. IN PATH TABLE (2 BYTES FOR NEW PTHPTR VALUE)  
 32 ; 81 = LOAD PC TYPE IMMEDIATE W/ adr.  
 33 ; 82 = CANCEL ALWAYS  
 34 ; 83 = CALL SUBROUTINE PATH (NEXT 2 BYTES ARE ADDRESS OF SUBROUTINE)  
 35 ; 84 = RETURN TO ORIGINAL PATH  
 36 ; 85 = LOAD SPEED DIRECT  
 37 ; 86 = CHECK BUCK BETWEEN POLES  
 38 ; 87 = MOTHER ZORBA TURNS AROUND  
 39 ;  
 40 ; ENTER:  
 41 ;  
 42 ; Y=OBJTAB INDEX FOR OBJECT TO MOVE  
 43 ;  
 44 ; EXIT:  
 45 ;  
 46 ; 00-02 = USED  
 47 ; X,Y,Z,S UPDATED  
 48 ;  
 49 ; PTHPTR POINTS TO NEXT DIRECTION BYTE. REPEAT COUNT UPDATED  
 50 ; IN OBJECT DESCRIPTOR TABLE. Z AND SPEED ARE UPDATED AS REQUIRED.  
 51 ;  
 52 ; IF A=0, STILL IN DISPLAY AREA.  
 53 ; IF A=FE, MOVED OUT OF DISPLAY AREA.  
 54 ;

7008	NXTRPT			
7008	20BB73	JSR	INDRPT	; INC POINTER TO NEXT REPEAT COUNT
7008	NXTR10			
700B	A120	LDA	[PTHPTR,X]	; GET NEW REPEAT COUNT
700D	3003 ^7012	BMI	NXTR20	; SPECIAL CONFIGURATION BYTE

57 700B A120 NXTR10  
700D 3003 ^7012 LDA [PTHPTR, X] ; GET NEW REPEAT COUNT  
BMI NXTR20 ; SPECIAL CONFIGURATION BYTE

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 61  
J:5200 .A65

NEXT REPEAT COUNT GETTER

700F 4C7E70 IMP NXTR70 ; GO TO Z SPEED AND EXIT  
7012 297F NXTR20 AND #FF ; SET UP FOR TABLE JUMP  
7012 297F CMP #8 ; SEE IF ANYTHING INVALID  
7014 C908  
7016 B0FA ^7012 BCS NXTR20  
7018 0A ASL A  
7019 A8 TAY  
701A B92970 LDA CONFGT, Y  
701D 8500 STA 0  
701E B92A70 LDA CONFGT+1, Y  
7022 8501 STA 1 ; RESTORE Y  
7024 A40A LDY 10  
7026 600000 JMP F01  
7029 CONFGT  
7029 4E70 DW NXTR40 ; JUMP  
702B 4570 DW NXTR80 ; LOAD PC IMMEDIATE  
702D 3970 DW NXTR21 ; CANCEL  
702E 5E70 DW NXTR50 ; CALL  
7031 6B70 DW NXTR60 ; RETURN  
7033 3C70 DW NXTR25 ; SET SPEED  
7035 D870 DW NXTR40 ; CHECK BUCK BETWEEN POLES & DEC UFOCNT  
7037 2571 DW NXTR10 ; MOTHER ZORBA TURN AROUND  
7039 NXTR21  
7039 A9EF LDA #\$FF ; CANCEL THIS ONE  
703B 60 RTS  
703C NXTR25  
703C 20B673 JSR GETNXT ; SET SPEED  
703F 990504 STA OBJJBL+5, Y  
7042 4C0B70 JMP NXTR10  
7045 NXTR30  
7045 20B673 JSR GETNXT ; GET ARGUMENT  
7048 990006 STA OBJJBL, Y ; SET TYPE  
704B 4C0B70 JMP NXTR10 ; NEXT  
704E NXTR40  
704E 20B673 JSR GETNXT ; POINT TO ARGUMENT  
7051 8500 STA 0 ; SET NEW PTHPTR LOW BYTE  
7053 A120 LDA PTHPTR, X ; GET NEW PTHPTR HI BYTE  
7055 9521 STA PTHPTR+1, X ; SET NEW PTHPTR HIGH BYTE  
7057 A500 LDA 0 ; SET NEW PTHPTR LOW BYTE  
7059 9520 STA PTHPTR, X ; ALL SET FOR NEXT  
705B 4C0B70 JMP NXTR10  
705E B520  
7060 9D0008 STA PTHPTR, X ; SAVE ADDRESS  
7063 B521 LDA RTNADD, X  
7065 9D0108 STA PTHPTR+1, X  
7068 4C4E70 JMP RTNADD+1, X ; JUMP TO NEW ADDRESS  
705F NXTR50  
705F 8500 LDA PTHPTR, X ; SAVE ADDRESS  
7060 9D0008 STA RTNADD, X  
7063 B521 LDA PTHPTR+1, X  
7065 9D0108 STA RTNADD+1, X  
7068 4C4E70 JMP NXTR40

1 RETURN TO PATH TABLE - RESTORES ADDRESS, POINTS TO NEXT ENTRY  
 2 AND CONTINUES

```

 4 706B      NXTR60
 5 706B  B00008  LDA    RTNADD,X ; RESTORE ADDRESS
 6 706E  9520   STA    PTHPTR,X
 7 7070  BD0108  LDA    RTNADD+1,X
 8 7073  9521   STA    PTHPTR+1,X
 9 7075  20BB73  JSR    INDRPT   ; SKIP OVER CALL INSTRUCTION
10 7078  20BB73  JSR    INDRPT   ; SKIP OVER ARG1
11 707B  4C0870  JMP    NXTRPT  ; EXECUTE NEXT COMMAND SKIPPING ARG2
12
13 ; MOVE Z, SPEED AND EXIT
14 ;
15 707E      NXTR70
16 707E  990606  STA    OBJTBL+6,Y ; SET REPEAT COUNT
17 7081  20BB73  JSR    INDRPT   ; INC POINTER TO NEXT DIRECTION BYTE
18 7084  A120   LDA    [PTHPTR,X] ; GET DIR BYTE TO 1
19 7086  8501   STA    1
20 7088  2920   AND    #$20    ; SEE IF TO MOVE IN Z
21 708A  F021 ^70AD  BEQ    NXTR90 ; NO
22 708C  A902   LDA    #$2    ; SEE IF POS/NEG
23 708E  2501   AND    1
24 7090  F010 ^70A2  BEQ    NXTR90 ; NEG
25 7092  B90306  LDA    OBJTBL+3,Y ; POS - INC Z
26 7095  18     CLC
27 7096  6201   ADC    #1
28 7098  C90A   CMP    #10    ; 9 IS MAX
29 709A  B011 ^70AD  BCS    NXTR90 ; TOO HIGH - LEAVE IT AND DO SPEED
30 709C  990306  STA    OBJTBL+3,Y
31 709E  4C4D70  JMP    NXTR90
32 70A2      NXTR90
33 70A2  B90306  LDA    OBJTBL+3,Y ; DEC Z
34 70A5  38     SEC
35 70A6  E901   SBC    #1
36 70A8  3003 ^70AD  BMJ    NXTR90 ; LOWER LIMIT EXCEEDED - LEAVE IT
37 70AA  990306  STA    OBJTBL+3,Y
38
39 ; NOW DO SPEED
40
41 70AD      NXTR90
42 70AD  A910   LDA    #$10    ; SEE IF TO MOVE IN SPEED
43 70AF  2501   AND    1
44 70B1  F013 ^70C6  BEQ    NXT110 ; NO - DONE
45 70B3  A901   LDA    #1    ; SEE IF POS/NEG
46 70B5  2501   AND    1
47 70B7  F010 ^70C9  BEQ    NXT120 ; NEG
48 70B9  B90506  LDA    OBJTBL+5,Y ; ADD TO SPEED
49 70BC  18     CLC
50 70BD  6908   ABC    #8
51 70BF  9002 ^70C3  BCC    NXT100
52 70C1  A9FF   LDA    #$FF    ; MAX AT FF
53 70C3      NXT100
54 70C3  990506  STA    OBJTBL+5,Y
55 70C6      NXT110
56 70C6  A900   LDA    #0    ; OK
57 70C8  60     RTS
58
59 70C9      NXT120
60 70C9  B90506  LDA    OBJTBL+5,Y ; NEG

```

57 70C6 A900 LDA #0 ; OK  
 70C8 60 RTS ; DONE  
 70C9 NXT120  
 70C9 B90506 LDA OBJTBL+5,Y ; NEG  
 58

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 63  
 NEXT REPEAT COUNT GETTER J:5200 .A65

70CC 38 SEC  
 70CB E908 SBC #8 ; SUBTRACT FROM SPEED  
 70CF F002 ^70D3 BCD NXT130 ; ZERO NO GOOD  
 70D1 B0E0 ^70C3 BCS NXT100  
 70D3 NXT130  
 70D3 A901 LDA #1 ; MINIMUM AT 1  
 70D5 40C370 JMP NXT100  
 70D8 ; GOT CONFIGURATION COMMAND TO CHECK BUCK BETWEEN POLES.  
 70D8 A580 LDA POLEND ; SEE IF THIS IS FIRST POLE CANCELED  
 70DA D00A ^70E6 BNE NXT150 ; NO - SEE IF BUCK BETWEEN THEM  
 70DC B90106 LDA OBJTBL+1,Y ; SET POLEND TO CONTAIN THIS POLE'S X  
 70DE 0901 DRA #1 ; FORCE NON-ZERO - ERROR NEGIGIBLE  
 70E1 8580 STA POLEND  
 70E3 4C0870 JMP NXTRPT  
 70E6 NXT150  
 70E6 A59A LDA BUCKX ; COMPARE TO SEE IF BUCK BETWEEN  
 70E8 18 CLC ; NORMALIZE TO FIND CENTER OF BUCK  
 70E9 6908 ADC #8  
 70EB D90106 CMP OBJTBL+1,Y ; BUCKX-THIS POLE X  
 70EE 9007 ^70F7 BCC NXT160 ; BUCKX < THIS POLE X  
 70F0 C580 CMP POLEND ; BUCKX > THIS POLE X  
 70F2 9007 ^70FB BCC NXT170 ; BUCKX < OTHER POLE - BETWEEN THEM  
 70F4 4C1E71 JMP NXT180 ; BUCKX > OLD POLE, TOO - NOT BETWEEN  
 70F7 NXT160  
 70F7 C580 CMP POLEND  
 70F9 9023 ^711E BCC NXT180 ; BUCKX < OLD POLE, TOO - NOT BETWEEN  
 70FB ; NXT170  
 70FB A900 LDA #0 ; BETWEEN POLES - BONUS 500PTS  
 70FD B600 STX 0 ; SAVE X  
 70FF A205 LDX #5  
 7101 208273 JSR ADDSCR  
 7104 A600 LDX 0 ; RESTORE X  
 7106 C6A6 DEC UEOCNT ; DEC UEO COUNT WHEN PASSING THRU POLES  
 7108 D004 ^710E BNE NXT175  
 710A A9FF LDA #0FF  
 710C 857B STA LVLFLG  
 710E 98 TYA ; SAVE X, Y DESTROYED BY UEODSP  
 710F 48 PHA  
 7110 8A TXA  
 7111 48 PHA  
 7112 208C60 JSR UEODSP  
 7115 A909 LDA #9  
 7117 20A843 JSR SNDINI  
 711A 68 PLA ; RESTORE X, Y  
 711B AA TAX  
 711C 68 PLA  
 711D A8 TAY  
 711E ; NXT180  
 711F A900 LDA #0 ; RESET POLEND  
 7120 8580 STA POLEND  
 7122 4C0870 JMP NXTRPT ; NOW GET NEXT REPEAT COUNT

1 MOTHER SHIP TURNS AROUND. GIVE IT ONE OF 2 OF THE SAUCER  
2 RETURN PATHS.

3  
4 7125 NXT190  
5 7125 A00AE8 LDA RANDOM ; SEE WHICH ONE TO USE  
6 7128 3005 ^712F BMI NXT200 ; USE #17  
7 712A A912 LDA #18 ; USE #18  
8 712C 4C3171 JMP NXT210  
9 712E NXT200...  
10 712F A911 LDA #17  
11 7131 NXT210  
12 7131 200FAD JSR SETPTH  
13 7134 B90706 LDA #BUTBL+7,Y ; SET ATTACK BIT  
14 7137 0910 DRA #\$10  
15 7139 990704 STA #BJTBL+7,Y  
16 713C 4C0870 JMP NXTRPT ; NOW MOVE ZORBA BACK  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

## UTILITIES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 65  
J:5200 ,A65

1 ; ROUTINE TO DISPLAY SCORE  
2 ; -----> UPDATES SCORE IN DISPLAY FROM CSCORE+3 TO CSCORE WITH  
3 ; CSCORE BEING MSB AND CSCORE+3 BEING LSB. THIS IS BACKWARDS FROM NORMAL.  
4  
5

6 713F SCRUDP  
7 713F A200 LDX #\$0  
8 7141 A000 LDY #\$0  
9 7143 B00E09 LDA CSCORE+1,X  
10 7146 4A LSR A  
11 7147 4A LSR A  
12 7148 4A LSR A  
13 7149 4A LSR A  
14 714A 18 CLC  
15 714B 6941 ADC #\$41  
16 714D 992F20 STA \$202F,Y  
17 7150 C8 INY  
18 7151 B00E09 LDA CSCORE+1,X  
19 7154 280F AND #\$0F  
20 7156 18 CLC  
21 7157 6941 ADC #\$41  
22 7159 992F20 STA \$202F,Y  
23 715C C8 INY  
24 715D E8 INX  
25 715E E003 CPX #\$3  
26 7160 D0E1 ^7143 BNE SCRUDP  
27 7162 60 RTS  
28  
29 ; ROUTINE TO DISPLAY TITLE PAGE  
30  
31 ; -----> DISPLAYS COPYRIGHT, HIGH SCORE (FROM HISCOR+3 TO HISCOR WITH  
32 ; HISCOR BEING MSB AND HISCOR+3 BEING LSB. BACKWARDS.  
33  
34 7163 TITLE  
35 7163 A900 LDA #0  
36 7165 8D00D4 STA \$D400  
37 7168 A946 LDA #LOW TLIST  
38 716A 8D0009 STA \$DLSTL  
39 716D A973 LDA #HIGH TLIST  
40 716E 8D0109 STA \$DHSTH  
41 7172 A9E8 LDA #\$E8 :@@@  
42 7174 8D0309 STA CHRAS  
43 7177 A940 LDA #\$40  
44 7179 8D0ED4 STA NMEN  
45 717C A922 LDA #Z00100010  
46 717E 8D0209 STA \$DMCTL  
47 7181 A920 LDA #\$20  
48 7183 8501 STA 1  
49 7185 A900 ZXCLB LDA #\$0  
50 7187 8500 STA 0  
51 7189 A8 TAY  
52 718A 9100 ZXCLP STA LO,Y  
53 718C 88 DEY  
54 718D D0F8 ^718A BNE ZXCLP  
55 718F E601 INC 1  
56 7191 A501 LDA 1  
57 7193 C940 CMP #\$40  
58 7195 D0EE ^7185 BNE ZXCLB  
59 7197 A200 LDX #\$0

UTILITIES

```

1 7199 B0BA72 OPTLOP LDA BR1,X
2 719A 38 SEC
3 719B E920 SBC #$20
4 719C 9F0020 STA $2000,X
5 71A2 E8 INX
6 71A3 F0C8 CPX #200
7 71A5 D0E2 ^7199 BNE OPTLOP
8 71A7 A200 LDX #$0
9 71A9 8E1509 STX COLOR4
10 71AC A000 LDY #$0
11 71AE B0QA09 HSCOUT LDA HISCOR+1,X
12 71B1 4A LSR A
13 71B2 4A LSR A
14 71B3 4A LSR A
15 71B4 4A LSR A
16 71B5 18 CLC
17 71B6 6920 ADC #$20
18 71B8 990E20 STA $200E,Y
19 71B9 C8 INY
20 71BC B0AO09 LDA HISCOR+1,X
21 71BF 290F AND #$0F
22 71C1 18 CLC
23 71C2 6920 ADC #$20
24 71C4 990E20 STA $200E,Y
25 71C7 C8 INY
26 71C8 E8 INX
27 71C9 E003 CPX #3
28 71CB D0E1 ^71AE BNE HSCOUT
29 71CD A200 LDX #$0
30 71CE A000 LDY #$0
31 71D1 B0OE09 RSCOUT LDA CSCORE+1,X
32 71D4 4A LSR A
33 71D5 4A LSR A
34 71D6 4A LSR A
35 71D7 4A LSR A
36 71D8 18 CLC
37 71D9 6920 ADC #$20
38 71DE 990320 STA $2003,Y
39 71DF C8 INY
40 71E0 B0OE09 LDA CSCORE+1,X
41 71E2 290F AND #$0F
42 71E4 18 CLC
43 71E5 6920 ADC #$20
44 71E7 990320 STA $2003,Y
45 71EA C8 INY
46 71EB E8 INX
47 71EC E003 CPX #3
48 71EE D0E1 ^71D1 BNE RSCOUT
49 71F0 A968 LDA #$48
50 71F2 9D1109 STA COLOR0
51 71F5 A990 LDA #$90
52 71F7 8D1209 STA COLOR1
53 71FA A9A8 LDA #$AB
54 71FC 8D1402 STA COLOR3
55 71FF A968 LDA #$48
56 7201 8D1309 STA COLOR2
57 7204 AD09E8 LDA KBCODE
58 7207 85AC STA SYSTAT
59 7209 AD0BD4 RBOW LDA VCOUNT
60 720C 18 CLC

```

```

1 720D 6578 ADC TIMER
2 720E 29E0 AND #11110000
3 7211 0908 ORA #000001000
4 7212 810A04 STA WSYNC
5 7216 8D18C0 STA $C018
6 7219 A578 LDA TIMER
7 721B 8D1409 STA COLOR3
8 721E A009E8 LDA KRCODE
9 7221 C5AC CMP SYSTAT
10 7223 D006 ^722B BNE CHKITA
11 7225 A010C0 LDA $C010
12 7228 D00E ^7209 BNE RBOW
13 722A 60 EXITR RTS
14 722B 29DF CHKITA AND #11011111
15 722D C919 CMP #700011001
16 722E D008 ^7209 BNE RBOW
17 7231 60 RTS
18
19 ; PAINT_BUCKSHIP/SHADOW
20
21 7232 A9C5 BSERAS LDA #LOW_SHADOW
22 7234 856B STA ADDR
23 7236 A996 LDA #HIGH_SHADOW
24 7238 856C STA ADDR+1
25 U 723A A900 LDA #LOW_MSHADOW
26 723C 856D STA ADDR
27 JI 723E A900 LDA #HIGH_MSHADOW
28 7240 856E STA ADDR+1
29 7242 A59A LDA BUCKX
30 7244 851D STA $1D
31 7246 A900 LDA #131
32 7248 851E STA $1E
33 724A 201868 JSR ERASE
34 724D A480 LDX LASTBS
35 724F B0A722 LDA BUKSL,X
36 7252 856B STA ADDR
37 7254 B0B172 LDA BUKSH,X
38 7257 856C STA ADDR+1
39 7259 B0B472 LDA MBUKSL,X
40 725C 856D STA ADDR
41 725F B0B772 LDA MRUKSH,X
42 7261 856E STA ADDR+1
43 7263 A59A LDA BUCKX
44 7265 851D STA $1U
45 7267 A59B LDA BUCKY
46 7269 851E STA $1E
47 726B 201868 JSR ERASE
48 726E 60 RTS
49 726F A9C5 RSEAIN LDA #LOW_SHADOW
50 7271 856B STA ADDR
51 7273 A996 LDA #HIGH_SHADOW
52 7275 856C STA ADDR+1
53 U 7277 A900 LDA #LOW_MSHADOW
54 7279 856D STA ADDR
55 JI 727B A900 LDA #HIGH_MSHADOW
56 727D 856E STA ADDR+1
57 727F A59A LDA BUCKX

```

```

1 7281 851D STA $1D
2 7282 A983 LDA #101
3 7285 851E STA $1E
4 7287 205A67 JSR PAINT
5 728A A480 LDX LASTBS
6 728C B1AE72 LDA BUKSL,X
7 728F 856B STA ADDR
8 7291 B0B172 LDA BUKSH,X
9 7294 856C STA ADDR+1
10 7296 B0B472 LDA MBUKSL,X
11 7299 856D STA ADDR
12 729B B0B772 LDA MBUKSH,X
13 729E 856E STA ADDR+1
14 72A0 A59A LDA BUCKY
15 72A2 851D STA $1D
16 72A4 A59B LDA BUCKY
17 72A6 851F STA $1E
18 72A8 205A67 JSR PAINT
19 72AB 4C9C6C JMP BSFIRE
20
21 THIS IS A TABLE TO DECODE BUCK ADDRESSES
22
23 72AE A79DB1 BUKSL DB LOW BUCKL,LOW BUCK,LOW BUCKR
24 72B1 269696 BUKSH DB HIGH BUCKL,HIGH BUCK,HIGH BUCKR
25
26 U 72B4 000000 MBUKSL DB LOW MBUCKL,LOW MBUCK,LOW MBUCKR
27 U 72B7 000000 MBUKSH DB HIGH MBUCKL,HIGH MBUCK,HIGH MBUCKR
28
29 72B8 53433A BR1 DB SCI
30 72BD R0R0R0R0R0 DB $B0,$B0,$B0,$B0,$B0,$B0
31 72C3 202048493A DB HI
32 72C8 B0B0B0B0B0 DB $B0,$B0,$B0,$B0,$B0,$B0
33 72CE 2020202020 DB
34 72D6 C2D5C3CB DB R'+128, 'U'+128, 'C'+128, 'K'+128
35 72DA 2020202020 DB
36 72E2 2020202020 DB
37 72E9 D2CFC7C5D2 DB 'R'+128, 'O'+128, 'G'+128, 'E'+128, 'R'+128, 'S'+128
38 72EF 2020202020 DB
39 72F6 202020434F DB COPYRIGHT 1983
40 730A 2020 DB
41 730C D3C5C7C1 DB 'S'+128, 'E'+128, 'G'+128, 'A'+128
42 7310 20454E5445 DB ENTERPRISES
43 731E 2020202020 DB INC.
44 7332 5052455353 DB 'PRESS
45 7338 1314011214 DB $13,$14,$01,$12,$14 ; 'START' FLASHING
46 733D 20544F2042 DB 'TO BEGIN'
47
48 7346 707070 TLIST DB $70,$70,$70
49 7349 4600207070 DB $46,$00,$20,$70,$70,$70,$07,$07,$70,$70,$50,$06,$20,$06,$20
50 7359 0670707070 DB $06,$70,$70,$70,$70,$70,$70,$70,$06
51 7362 41 DB $41
52 7363 4673 DW TLIST
53
54 ; CHECK FOR NEW SCORE HIGHER THAN OLD SCORE
55 ;
56
57 7365 A200 LDX #0
58 7367 CHKHSC

```

7347 8P0P09 LDA CSCORE,X  
 7348 0D0909 CMP HISCOR,X  
 7349 9012 7381 BCC CHKH40  
 734F D005 7376 BNE CHKH20  
 7371 F8 TXN  
 7372 F004 CPX #4  
 7374 10F1 ^7367 BNE CHKH10  
 7375 60 BPL CHKH20  
 7376 A203 LDX #3  
 7378 60 BPL CHKH30  
 7378 8P0P09 LDA CSCORE,X  
 737B 9D0909 STA HISCOR,X  
 737E CA DEX  
 737F 10F7 ^7378 BPL CHKH50  
 7381 60 RTS  
 7382 : INCREMENT SOUND POINTER  
 7382 F662 INSEPT INC SNDPTR,X  
 7384 D002 ^7388 BNE INSD10  
 7386 F663 INC SNDPTR+1,X  
 7388 60 RTS  
 7388 : ROUTINE TO ADD NUMBER IN X,A (BCD, X IS MSB) TO SCORE  
 7389 ADDSCR  
 7389 18 CLC  
 738A F8 SED  
 738B 6D1009 ADC CSCORE+3  
 738E 8D1009 STA CSCORE+3  
 7391 8A TXA  
 7392 6D0E09 ADC CSCORE+2  
 7395 8D0E09 STA CSCORE+2  
 7398 A900 LDA #0  
 739A 6D0E99 ADC CSCORE+1  
 739D 8D0E09 STA CSCORE+1  
 73A0 A900 LDA #0  
 73A2 6D0P09 ADC CSCORE  
 73A5 8P0P09 STA CSCORE  
 73A8 B8 CLD  
 73A9 60 RTS  
 73A9 : ADD 16 TO REGISTER PAIR [5,6]  
 73AA ADD1625  
 73AA A505 LDA 5  
 73AC 18 CLC  
 73AD 6910 ADC #16  
 73AE 8505 STA 5  
 73B1 9002 ^73B5 BCC ADD810  
 73B3 F606 INC 6  
 73B5 ADD810  
 73B5 60 RTS  
 73B5 : ROUTINE FOR GETTING NEXT BYTE FROM PATH TABLE AND POINTING  
 : TO BYTE FOLLOWING THAT

```

1 73B4 20BB73  DETNXT
2 73B4 20BB73  JSR INDRPT
3 73B9 A120  LDA [PTHPTR,X]
4
5 ; ROUTINE FOR INCREMENTING WORD AT PTHPTR+X.
6
7 73BB INDRPT
8 73BB F620  INC PTHPTR,X
9 73BD D002 ^73C1  BNE INDR10
10 73BF F621  INC PTHPTR+1,X
11 73C1 INDR10
12 73C1 60  RTS
13
14 ; MULTIPLY ROUTINE MULTIPLIES TO 8 BIT NUMBERS IN A,X
15 ; AND CREATES A 16 BITE PRODUCT IN A,X. A CHECK IS
16 ; MADE TO ASSURE THAT MULTIPLICAND IS LESS THAN MULTIPLIER
17 ; SO THAT THE ROUTINE EXECUTES AT MAXIMUM SPEED.
18
19 ; ENTER:
20 ; X=MULTPLICAND
21 ; A=MULTIPLIER
22
23 ; EXIT:
24 ; 00-02 = USED
25 ; A = RESULT HIGH
26 ; X = RESULT LOW
27
28
29 73C2 MULT
30 73C2 E000  CPX #0 ; TEST IF MULTPLICAND=0
31 73C4 F017 ^73D0  BEQ MULT30 ; DEC MULTPLICAND TO AVOID
32 73C6 CA  DEX ; THE CLR BEFORE ADC 0
33 73C7 8400  STX 0 ; SHIFT OUT LOW BIT FOR
34 73C9 4A  LSR A ; THE FIRST ITERATION
35 73CA 8501  STA 1 ; RESULT HI=0
36 73CC A900  LDA #0 ; 8 BITS=8 ITERATIONS
37 73CE A208  LDX #8 ; TEST LO BIT OF MULTIPLIER
38 73D0 9002 ^73D4  BCC MULT20 ; ADD IN TO PRODUCT
39 73D2 6500  ADC 0 ; SHIFT RESULT DOWN 1 BIT AND
40 73D4 6A  ROR A ; SHIFT OUT NEXT BIT OF MULTIPLIER
41 73D5 6601  RDR 1 ; SET LSB RESULT
42 73D7 CA  DEX
43 73D8 D0E6 ^73D0  BNE MULT10
44 73D9 A601  LDX 1 ; COPY RAM COLORS TO HARDWARE REGS.
45 73D9 60  RTS
46
47 73D0 8A  TXA
48 73D0 60  RTS
49
50
51
52 73D9 VRIRTN
53 73D9 48  PHA
54 73E0 AD1109  LDA COLOR0
55 73E3 8D16C0  STA $C016
56 73E6 AB1209  LDA COLOR1
57 73E9 8D17C0  STA $C017
58
59 73EC AD1309  LDA COLOR2
60 73EF 8D18C0  STA $C018

```

```

1 73E2 AD1409 LDA COLOR3
2 73E5 801900 STA $0019
3 73E8 AD1509 LDA COLOR4
4 73EB 801A00 STA $001A
5 73FF A204 LDA ##4
6 7400 801E00 STA CONSOL ; RESET CONSOL SWITCH
7 7403 AD0009 LDA SDLSHL
8 7406 800204 STA $0402
9 7409 AD0109 LDA SDLSHT
10 740C 800304 STA $0403
11 740F AD0209 LDA SDMCTL
12 7412 800004 STA DMACTL
13 7415 AD00E8 LDA $E800
14 7418 801609 STA PDLO
15 741B AD01E8 LDA $E801
16 741E 801709 STA PDLY
17 7421 800B08 STA $E80B
18 7424 AD0409 LDA CHART
19 7427 800104 STA CHACTL
20 742A AD0309 LDA CHBAS
21 742D 800904 STA CHBASE
22 7430 E678 TIC TIMER
23 7432 68 PLA
24 7433 40 RTI

```

; DO JOYSTICK WORK

; READS JOYSTICK AND PLACES DATA IN JSDATA.  
; DATA IS AS FOLLOWS:

; BIT 0-  
; BIT 1-  
; BIT 2- UP  
; BIT 3- DOWN  
; BIT 4- LEFT  
; BIT 5- FIRE  
; BIT 6-  
; BIT 7- RIGHT

```

45 7434 JOYSTK
46 7434 A9FF LDA ##FF
47 7436 5560 STA JSDATA
48 7438 A59C LDA FULAMT ; SEE IF OUT OF FUEL
49 743A 059D ORA FULAMT+1
50 743C 059E ORA FULAMT+2
51 743E D003 >7443 BNE JOYS10
52 7440 4C9074 JMP JOYS90
53 7443 AD1409 LDA PDLO
54 7446 C933 CMP #51
55 7448 B009 ^7453 BCS JS1
56 744A A560 LDA JSDATA
57 744C 29EF AND #11101111
58 744E 6560 STA JSDATA
59 7450 4C5D74 JMP YAXIS
60 7453 C9B2 JS1 CMP #178

```

UTILITIES

```

1 7455 9006 ^7450      BCC      YAXIS
2 7457 A560      LDA      JSADATA
3 7459 297F      AND      #%01111111
4 745B 8560      STA      JSADATA
5 745D AD1709      YAXIS    LDA      PDL1
6 7460 C933      CMP      #51
7 7462 B009 ^746D      BCS      JS2
8 7464 A560      LDA      JSADATA
9 7466 29FB      AND      #%11111011
10 7468 8560      STA      JSADATA
11 746A 4C7774      JMP      JSJAM
12 746D C9B2      JS2      CMP      #178
13 746F 9006 ^7477      BCC      JSJAM
14 7471 A560      LDA      JSADATA
15 7473 29F7      AND      #%11110111
16 7475 8560      STA      JSADATA
17 7477 AD0EE8      JSJAM    LDA      SKSTAT
18 747A 2908      AND      #%00001000
19 747C D006 ^7484      BNE      JSJAM1
20 747E A560      LDA      JSADATA
21 7480 29DF      AND      #%11011111
22 7482 8560      STA      JSADATA
23 7484 AD10C0      JSJAM1   LDA      $C010
24 7487 D006 ^748E      BNE      JSJAM2
25 7489 A560      LDA      JSADATA
26 748B 29DF      AND      #%11011111
27 748D 8560      STA      JSADATA
28 748F 60      JSJAM2    RTS
29 7490 A9F7      LDA      #$F7
30 7492 8560      STA      JSADATA      ; FIRCE DOWN WHEN OUT OF FUEL
31 7494 A59B      LDA      BUCKY
32 7496 C978      CMP      #120      ; SEE IF ON GROUND
33 7498 B001 ^749B      BCS      JOY100      ; TIME TO DIE
34 749A 60      RTS
35 749B JOY100
36 749B 4C6765      JMP      DEADBK
37
38
39
40
41
42
43 ; ROUTINE FOR INC/DEC WORD AT 0,1 AND 2,3
44
45 749E INC01
46 749E E600      INC      0
47 74A0 D002 ^74A4      BNE      INC010
48 74A2 E601      INC      1
49 74A4 INC010
50 74A4 A501      LDA      1      ; GET MSB TO A
51 74A6 60      RTS
52 74A7 INC23
53 74A7 E602      INC      2
54 74A9 D002 ^74A0      BNE      INC210
55 74AB E603      INC      3
56 74AD INC210
57 74AD A503      LDA      3      ; MSB TILA
58 74AF 60      RTS
59 LIST      L

```

## 1 : SOUND TABLES

2 SN0TBL

3 74B0	4 0875	5 DW	6 CRASH-1	7 ; ALIEN DIED	
5 74B2	6 3575	7 DW	8 BFIREF-1	9 ; BUCK FIRE	
7 74B4	8 DC74	9 DW	10 BXPL-1	11 ; BUCK DIED	
9 74B6	10 5175	11 DW	12 HFIRE-1	13 ; HOPPER FIRE	
11 74B8	12 5E75	13 DW	14 PFIRE-1	15 ; POLE FIRE	
13 74BA	14 2775	15 DW	16 RICOCH-1	17 ; RICOCHET SOUND (SHOT BOUNCE OFF ZORBA)	
15 74BC	16 6B75	17 DW	18 FUELST-1	19 ; FUEL OUT	
17 74BE	18 4375	19 DW	20 RETRWN-1	21 ; WHITE NOISE W/ BUCK SHOT	
19 74C0	20 7675	21 DW	22 ZORRBD-1	23 ; ZORBA DEAD	
21 74C2	22 B375	23 DW	24 PXLPAS-1	25 ; PASS THRU POLE	
23 74C4	24 8574	25 DW	26 BONMS-1	27 ; FREESHIP	
25 74C6	26 01AE303000	27 BONUS	28 DB	29 1, \$AE, \$30, \$30, \$0, \$30, \$30, \$0, \$30, \$30, \$0	
27	28	29	30	31	
30 74D1	31 0203030104	32 BRECHN	33 DB	34 2, 3, 3, 1, 1, 1, 2, 3, 2, 2, 2, 2	
32	33	34	35	36	
34 74D0	35	36 BEXPL	37	38	
36 74D0	37 018E80EE10	38 DB	39 1, \$8F, \$80, \$FF, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58		
38 74EB	39 6068707880	40 DB	41 \$60, \$68, \$70, \$78, \$80, \$88, \$90, \$98, \$A0, 1, \$8D, \$A8, \$B0, \$B8, 1, \$8B, \$C0, \$CB		
40 74FTL	41 D00189D801	42 DB	43 \$D0, 1, \$89, \$D8, 1, \$87, \$E0, 1, \$85, \$E8, 1, \$83, \$F0, \$FF, 0		
42 750C	43	44 CRASH	45	46	
44 750C	45 018FF0E040	46 DB	47 1, \$8F, \$F0, \$E0, \$40, \$50, \$60, \$40, \$70, \$80, \$90, \$A0, \$B0, \$C0		
46 751A	47 95018A8060	48 DB	49 \$95, \$1, \$8A, \$80, \$60, \$55, \$40, \$45, \$30, \$35, \$25, \$20, \$15, \$0		
48	49	50	51	52	
50 7528	51	52 RICOCH	53	54	
52 7528	53 01AA401004	54 DB	55 1, \$AA, \$40, \$10, \$04, \$20, \$28, \$30, \$38, \$40, \$48, \$58, \$70, 0		
54 7536	55	56 BFIREF	57	58	
56 7536	57 01AA808890	58 DB	59 1, \$AA, \$80, \$88, \$90, \$98, \$A0, \$A8, \$B0, \$B8, \$C0, \$CB, \$D0, 0		
58 7544	59	60 RETRWN	61	62	
60 7544	61 0184081018	62 DB	63 1, \$84, \$08, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58, 0		
62 7552	63	64 HEIRE	65	66	
64 7552	65 01AA101810	66 DB	67 1, \$AA, \$10, \$18, \$10, \$18, \$10, \$18, \$10, \$18, \$10, \$18, 0		
66 755F	67	68 PFIRE	69	70	
68 755F	69 01AA405240	70 DB	71 1, \$AA, \$40, \$52, \$40, \$52, \$40, \$52, \$40, \$52, \$40, \$52, 0		
70 756C	71	72 FUELST	73	74	
72 756C	73 01AA807870	74 DB	75 1, \$AA, \$80, \$78, \$70, \$68, \$60, \$58, \$50, \$48, 0		
74 7577	75	76 ZORRBD	77	78	
76 7577	77 018F80FF10	78 DB	79 1, \$8F, \$80, \$FF, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58		
78 7585	79 018F80FF10	80 DB	81 1, \$8F, \$80, \$FF, \$10, \$18, \$20, \$28, \$30, \$38, \$40, \$48, \$50, \$58		
80 7593	81 6068707880	82 DB	83 \$60, \$68, \$70, \$78, \$80, \$88, \$90, \$98, \$A0, 1, \$8D, \$A8, \$B0, \$B8, 1, \$8B, \$C0, \$CB		
82 75A5	83 D00189D801	84 DB	85 \$D0, 1, \$89, \$D8, 1, \$87, \$E0, 1, \$85, \$E8, 1, \$83, \$F0, \$FF, 0		
84 75B4	85	86 01841F101C	87 PXLPAS	88 DB	89 1, \$84, \$1F, \$1D, \$1C, \$1B, \$1A, \$10, \$F, \$F, \$D, \$C, \$B, \$A, \$9, \$8, \$7, \$6, \$5, 0
86	87	88	89	90	91

## 50 : PATH TABLES FOR OBJECTS MOVING

51 : SPECIAL CONFIGURATION COMMANDS ARE EQUATED FOR READING TABLE

52 = 0080	53 JUMP	54 EQU	55 \$80	56 ; JUMP IN TABLE
54 = 0081	55 LOADPC	56 EQU	57 \$81	58 ; LOAD PC TYPE IMMEDIATE
56 = 0082	57 KILL	58 EQU	59 \$82	60 ; CANCEL
58 = 0083	59 CALL	60 EQU	61 \$83	62 ; CALL IN TABLE (1 LEVEL)
60 = 0084	61 RET	62 EQU	63 \$84	64 ; RETURN FROM CALL
62 = 0085	63 SETSPD	64 EQU	65 \$85	66 ; SET SPEED

= 0086 EQU \$86 ; POLES FAST BUGH CHECK FOR BUCK BETWEEN  
 = 0087 EQU \$87 ; MOTHER SHIP TAKE RETURN PATH  
 ; TABLE POINTING TO PATHS

5 75C8 PTHTBL  
 6 75C8 1576 DW PATH0-1 ; BUCK SHOTS  
 7 75CA 2876 DW PATH1-1 ; LEFT POLES  
 8 75CC 6076 DW PATH2-1 ; RIGHT POLES  
 9 75CE 9A76 DW PATH3+1 ; SAUCER PATH#1  
 10 75D0 A276 DW PATH4+1 ; SAUCER PATH#2  
 11 75D2 A876 DW PATH5+1 ; SAUCER PATH#3  
 12 75D4 B276 DW PATH6+1 ; SAUCER PATH#4  
 13 75D6 B876 DW PATH7-1 ; EXPLOSION PATH  
 14 75D8 D276 DW PATH8-1 ; CANCEL OBJECT  
 15 75DA E076 DW PATH9-1 ; BUCK EXPLOSION PATH  
 16 75DC D376 DW PATH10-1 ; HOPPER PATHS 1-2  
 17 75DE F576 DW PATH11-1  
 18 75E0 1777 DW PATH12-1 ; ENEMY SHOT PATH  
 19 75E2 2477 DW PATH13-1 ; POLE RAY LEFT  
 20 75E4 2977 DW PATH14-1 ; POLE RAY RIGHT  
 21 75E6 2E77 DW PATH15-1 ; MOTHER ZORBA ENTRANCE PATH#1  
 22 75E8 6777 DW PATH16-1 ; MOTHER ZORBA ENTRANCE PATH#2  
 23 75EA A077 DW PATH17-1 ; MOTHER ZORBA RETURN PATH 1  
 24 75EC B777 DW PATH18-1 ; MOTHER ZORBA RETURN PATH 2  
 25 75EE CE77 DW PATH19-1 ; STAR PATHS 19-30  
 26 75F0 D377 DW PATH20-1  
 27 75F2 DA77 DW PATH21-1  
 28 75F4 E177 DW PATH22-1  
 29 75F6 E677 DW PATH23-1  
 30 75F8 ED77 DW PATH24-1  
 31 75FA F477 DW PATH25-1  
 32 75FC F977 DW PATH26-1  
 33 75FE 0078 DW PATH27-1  
 34 7600 0778 DW PATH28-1  
 35 7602 0C78 DW PATH29-1  
 36 7604 1378 DW PATH30-1  
 37 7606 1A78 DW PATH31-1 ; RICOCHET SHOTS #'S 31-38  
 38 7608 2978 DW PATH32-1  
 39 760A 3878 DW PATH33-1  
 40 760C 4278 DW PATH34-1  
 41 760E 5678 DW PATH35-1  
 42 7610 5B78 DW PATH36-1  
 43 7612 6A78 DW PATH37-1  
 44 7614 7978 DW PATH38-1

; PATH0 IS PATH FOR BUCK SHOTS  
 ;  
 48 7616 PATH0

49 7616 184083 DB 24,\$40, CALL  
 50 7617 2276 DW SUBO  
 51 7618 83 DB CALL  
 52 761C 2276 DW SUBO  
 53 761E 83 DB CALL  
 54 761F 2276 DW SUBO  
 55 7621 82 DB KILL  
 56 7622 SUBO DB 16,\$72,4,\$40,4,\$88,RET  
 57 7622 1072044004 DB

; PATH1, PATH2 FOR POLES

## ; PATH1, PATH2 FOR POLES

## DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 75  
J:5200 .A65

1629	PATH1	DB	2,\$64
7629	0264	DB	
762B	016401C403	DB	1,\$64,1,\$C4,3,\$44,1,\$C4
7633	83	DB	CALL
7634	B179	DW	SUB1
7634	83	DB	CALL
7637	B179	DW	SUB1
7639	83	DB	CALL
763A	B179	DW	SUB1
763C	0244	DB	2,\$44
763E	016401C483	DB	1,\$64,1,\$C4,CALL
7643	B879	DW	SUB3
7645	0244	DB	2,\$44
7647	014401C483	DB	1,\$44,1,\$C4,CALL
764C	R179	DW	SUB1
764E	83	DB	CALL
764F	BC79	DW	SUB4
7651	83	DB	CALL
7652	B179	DW	SUB1
7654	83	DB	CALL
7655	B879	DW	SUB3
7657	86	DB	POLPAS
7658	83	DB	CALL
7659	B179	DW	SUB1
765B	83	DB	CALL
765C	B879	DW	SUB3
765F	28C4R2	DB	40,\$C4,KILL
7661	PATH2	DB	2,\$44
7661	0264	DB	
7663	016401CC03	DB	1,\$64,1,\$CC,3,\$44,1,\$CC
766B	83	DB	CALL
766C	C179	DW	SUB5
766E	83	DB	CALL
766F	C179	DW	SUB5
7671	83	DB	CALL
7672	C179	DW	SUB5
7674	0244	DB	2,\$44
7676	016401CC83	DB	1,\$64,1,\$CC,CALL
767B	C879	DW	SUB7
767D	0244	DB	2,\$44
767F	014401CC83	DB	1,\$44,1,\$CC,CALL
7684	C179	DW	SUB5
7686	83	DB	CALL

## DATA TABLES

```

7687 CC79 DW SUB6
1 7689 83 DB CALL
2 768A 6179 DW SUB5
3
4 768C 83 DB CALL
5 768D CB79 DW SUB7
6
7 768F 86 DB POLPAS ; CHECK BUCK BETWEEN POLES
8 7690 83 DB CALL
9 7691 6179 DW SUB5
0 7693 83 DB CALL
1 7694 CB79 DW SUB7
2
3 7696 280082 DB 40,$C,KILL
4
5 : PATH 3-4 ARE SAUCERS
6
7 7699 PATH3
8 7699 7040 DB 112,$40
9 769B 83 DB CALL
10 769C 6179 DW SUB9 : ENTRANCE
11 769E 83 DB CALL
12 769F 7879 DW SUB10 : EXIT
13
14 76A1 PATH4
15 76A1 7040 DB 112,$40
16 76A3 83 DB CALL
17 76A4 6179 DW SUB9 : ENTRANCE
18 76A6 83 DB CALL
19 76A7 A079 DW SUB12 : EXIT
20
21 76A9 PATH5
22 76A9 7040 DB 112,$40
23 76AB 83 DB CALL
24 76AC 8979 DW SUB11 : ENTRANCE
25 76AF 83 DB CALL
26 76AF A079 DW SUB12 : EXIT
27
28 76B1 PATH6
29 76B1 7040 DB 112,$40
30 76B3 83 DB CALL
31 76B4 8979 DW SUB11 : ENTRANCE
32 76B6 83 DB CALL
33 76B7 7879 DW SUB10 : EXIT
34
35 76B9 PATH7
36 76B9 8107 DB LOADPC,7 SET AS EXPLOSION
37 76B9 8510 DB SETSPD,$10 ; SET SPEED
38 76BD 0500 DB 5,0 ; CYCLE THROUGH SIZES AND STOP
39 76BF 0520 DB 5,$20
40
41 76C1 PATH9
42 76C1 8510 DB SETSPD,$10
43 76C3 05A8 DB 5,$A8
44 76C5 0522 DB 5,$22
45 76C7 0522 DB 5,$22
46 76C9 05AA DB 5,$AA
47 76CB 0522 DB 5,$22
48 76CD 0522 DB 5,$22
49 76CF 0522 DB 5,$22
50
51
52
53
54
55
56
57

```

## DATA TABLES

76D1	0522	DB	5, \$22	
76D3	82	DB	KILL	CANCEL
; HOPPER PATHS				
76D4	PATH10			; PATH10 FOR BACK/FORTH LEFT TO RIGHT
76D4	04C404E401	DB	4, \$C4, 4, \$E4, 1, \$20, CALL	
76D8	5279	DW	SUB22	
76D0	10C4012083	DB	16, \$C4, 1, \$20, CALL	
76E2	C778	DW	SUB14	
76E4	10CC08EC01	DB	16, \$CC, 8, \$EC, 1, \$20, CALL	
76F8	1479	DW	SUB20	
76E8	20C4012083	DB	32, \$C4, 1, \$20, CALL	
76F2	8978	DW	SUB14	
76F4	7ECC	DB	127, \$CC	
76E6	PATH11			; BACKWARDIS PATH10
76F6	04CC04EC01	DB	4, \$CC, 4, \$EC, 1, \$20, CALL	
76FD	E678	DW	SUB17	
76FF	10CC012083	DB	16, \$C4, 1, \$20, CALL	
7704	3379	DW	SUB21	
7706	10C408E401	DB	16, \$C4, 8, \$E4, 1, \$20, CALL	
770D	A878	DW	SUB15	
770E	20CC012083	DB	32, \$CC, 1, \$20, CALL	
7714	E578	DW	SUB19	
7716	7FC4	DB	127, \$C4	
7718	PATH12			; ENEMY SHOTS
7718	83	DB	CALL	
7719	1E77	DW	SUB23	
771B	80	DB	JUMP	
771C	1877	DW	PATH12	
771E	04C4046408	DB	4, \$C4, 4, \$64, 8, \$64, RET	
7725	PATH13			; LEFT POLE RAY
7725	7E8080	DB	127, \$80, JUMP	
7728	2577	DW	PATH13	
772A	PATH14			; RIGHT POLE RAY
772A	7F8880	DB	127, \$88, JUMP	
772B	2A77	DW	PATH14	
772F	PATH15			; ZORBA ENTRANCE PATH #1
772E	0840	DB	8, \$40	
7731	10C8	DB	16, \$C8	
7733	10BA	DB	16, \$BA	
7735	0840	DB	8, \$40	
7737	08C0	DB	8, \$C0	
7739	4880	DB	72, \$80	
773B	0840	DB	8, \$40	
773D	10B2	DB	16, \$B2	
773F	0840	DB	8, \$40	
7741	08C8	DB	8, \$C8	
7743	48BA	DB	72, \$BA	
7745	08C8	DB	8, \$C8	
7747	0450	DB	4, \$50	
7742	04C0	DB	4, \$C0	
774B	10A2	DB	16, \$A2	

## DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 78  
J:5200 A65

774D	0200	DB	2,\$C0	
774F	02FA	DB	2,\$FA	
7751	0880	DB	8,\$80	
7753	02C8	DB	2,\$C8	
7755	0272	DB	2,\$72	
7757	0280	DB	2,\$80	
7759	02C0	DB	2,\$C0	
775B	02EA	DB	2,\$EA	
775D	0280	DB	2,\$80	
775E	02E2	DB	2,\$E2	
7761	0888	DB	8,\$88	
7763	0880	DB	8,\$80	
7765	0888	DB	8,\$88	
7767	87	DB	MTRBAK	; MOTHER ZORBA COME BACK NOW
7768	PATH16	DB	16,\$40	; ZORBA ENTRANCE PATH #2
7769	0840	DB	8,\$40	
776A	1000	DB	16,\$C0	
776C	10R2	DB	14,\$R2	
776E	0840	DB	8,\$40	
7770	08C8	DB	8,\$C8	
7772	4888	DB	72,\$88	
7774	0840	DB	8,\$40	
7776	10RA	DB	16,\$RA	
7778	0840	DB	8,\$40	
777A	08C0	DB	8,\$C0	
777C	48B2	DB	72,\$B2	
777E	08C0	DB	8,\$C0	
7780	0450	DB	4,\$50	
7782	04C8	DB	4,\$C8	
7784	10AA	DB	16,\$AA	
7786	02C8	DB	2,\$C8	
7788	02F2	DB	2,\$F2	
778A	0888	DB	8,\$88	
778C	02C0	DB	2,\$C0	
778E	0272	DB	2,\$72	
7790	0288	DB	2,\$88	
7792	02C8	DB	2,\$C8	
7794	02E2	DB	2,\$E2	
7796	0288	DB	2,\$88	
7798	02EA	DB	2,\$EA	
779A	0880	DB	8,\$80	
779C	0888	DB	8,\$88	
779E	0880	DB	8,\$80	
77A0	87	DB	MTRBAK	; MOTHER ZORBA COME BACK NOW

## DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 79  
J:5200 A6577A1 02E8 PATH17 ; MOTHER ZORBA RETURN PATH #1  
77A1 02E8 DB \$EC

77A3 10A0 DB 16,\$A0

77A5 04F0 DB 4,\$F0

77A7 1088 DB 16,\$88

77A9 04E4 DB 4,\$E4

77AB 2080 DB 32,\$80

77AD 08F0 DB 8,\$F0

77AF 20A8 DB 32,\$A8

77B1 08E4 DB 8,\$E4

77B3 40A0 DB 64,\$A0

77B5 7FDD DB 127,\$DD

77B7 82 DB KILL

77B8 02E8 PATH18 ; MOTHER ZORBA RETURN PATH #2

77B8 02E4 DB 2,\$E4

77BA 10A8 DB 16,\$A8

77BC 04F5 DB 4,\$F5

77BE 1080 DB 16,\$80

77C0 04EC DB 4,\$EC

77C2 2088 DB 32,\$88

77C4 08E5 DB 8,\$E5

77C6 20A0 DB 32,\$A0

77C8 08EC DB 8,\$EC

77CA 40A8 DB 64,\$A8

77CC 7F05 DB 127,\$05

77CE 82 DB KILL

; STAR PATHS 19-30

ZZCF PATH19

77CF 7E4080 DB 127,\$40,JUMP

77D2 CF77 DW PATH19

77D4 PATH20

77D4 014001C880 DB 1,\$40,1,\$C8,JUMP

77D9 D477 DW PATH20

77DB PATH21

77DB 018801C880 DB 1,\$88,1,\$C8,JUMP

77E0 DB77 DW PATH21

77E2 PATH22

ZZE2 7E8880 DB 127,\$88,JUMP

77E5 E277 DW PATH22

77E7 PATH23

; STRAIGHT UP

; RIGHT

## DATA TABLES

77E7	018801CC80	DB	1,\$88,1,\$CC,JUMP	
77E8	E777	DW	PATH23	
77E9	7777	PATH24	DB	1,\$44,1,\$CC,JUMP
77E3	EE77	DW	PATH24	
77E5	EE77	PATH25	DB	127,\$44,JUMP
77E5	7F4480	DB	127,\$44,JUMP	: DOWN
77F8	E577	DW	PATH25	
77FA	EE77	PATH26	DB	1,\$44,1,\$C4,JUMP
77EA	014401C480	DB	1,\$44,1,\$C4,JUMP	
77EF	EA77	DW	PATH26	
7801	EE77	PATH27	DB	1,\$80,1,\$04,JUMP
7801	018001C480	DB	1,\$80,1,\$04,JUMP	
7806	0178	DW	PATH27	
7808	EE77	PATH28	DB	127,\$80,JUMP
7809	7F8080	DB	127,\$80,JUMP	: LEFT
780B	0878	DW	PATH28	
780D	EE77	PATH29	DB	1,\$80,1,\$C0,JUMP
7812	0B78	DW	PATH29	
7814	EE77	PATH20	DB	1,\$40,1,\$C0,JUMP
7814	014001C080	DB	1,\$40,1,\$C0,JUMP	
7819	1478	DW	PATH30	
781B	EE77	PATH31	DB	1,\$80,1,\$C0,JUMP
781B	810483	DB	LOADPC,4,CALL	
781E	2378	DW	SUB24	
7820	80	DB	JUMP	
7821	1B78	DW	PATH31	
7823	EE77	SUB24	DB	4,\$C0,4,\$60,8,\$60,RET
7823	04C0046008	DB	4,\$C0,4,\$60,8,\$60,RET	
782A	EE77	PATH32	DB	4,\$C0,4,\$60,8,\$60,RET
782A	810483	DB	LOADPC,4,CALL	
782D	3278	DW	SUB25	
782F	80	DB	JUMP	
7830	2A78	DW	PATH32	
7832	EE77	SUB25	DB	4,\$C0,4,\$E8,8,\$E8,RET
7832	04C004E808	DB	4,\$C0,4,\$E8,8,\$E8,RET	
7839	EE77	PATH33	DB	LOADPC,4,CALL
7839	810483	DB	LOADPC,4,CALL	
783C	4178	DW	SUB26	
783E	80	DB	JUMP	
783F	3978	DW	PATH33	
7841	EE77	SUB24	DB	4,\$88,4,\$A8,8,\$A8,RET
7841	048804A808	DB	4,\$88,4,\$A8,8,\$A8,RET	
7843	EE77	PATH34	DB	LOADPC,4,CALL
7843	810483	DB	LOADPC,4,CALL	
784B	5078	DW	SUB27	
784D	80	DB	JUMP	
784F	4878	DW	PATH34	
7850	EE77	SUB27	DB	4,\$CC,4,\$EC,8,\$EC,RET
7850	04CC04EC08	DB	4,\$CC,4,\$EC,8,\$EC,RET	
7857	EE77	PATH35	DB	LOADPC,4,JUMP
7857	810480	DB	LOADPC,4,JUMP	
785A	1877	DW	PATH12	
785C	EE77	PATH36	DB	LOADPC,4,CALL
785C	810483	DB	LOADPC,4,CALL	
785F	6478	DW	SUB28	

785C 810483 DB LOADPC, 4, CALL  
785F 6478 DW SUB28

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 81  
J:5200 .A65

DATA TABLES

7861	80	DB	JUMP
7862	5C78	DW	PATH36
7864	SUB28	DB	4, \$C4, 4, \$E4, 8, \$E4, RET
7864	04C404E408	DB	
786B	810483	DB	LOADPC, 4, CALL
786E	7378	DW	SUB29
7870	80	DB	JUMP
7871	6B78	DW	PATH37
7873	SUB29	DB	4, \$80, 4, \$A0, 8, \$A0, RET
7873	048004A008	DB	
787A	PATH38	DB	
787A	810483	DB	LOADPC, 4, CALL
787D	8278	DW	SUB30
787F	80	DB	JUMP
7880	7A78	DW	PATH38
7882	SUB30	DB	4, \$C0, 4, \$E4, 8, \$E4, RET
7882	04C004E408	DB	
7889	SUB14	DB	
7889	04C80A4009	DB	4, \$C8, 10, \$40, 9, \$C8, 4, \$40, 11, \$C8, 3, \$88, 2, \$C8, 7, \$A8
7899	02C003880B	DB	2, \$CC, 3, \$88, 11, \$CC, 4, \$44, 9, \$CC, 10, \$44, 4, \$CC, RET
78A8	SUB15	DB	
78A8	03C8044005	DB	3, \$C8, 4, \$40, 5, \$C8, 4, \$40, 2, \$C8, 2, \$40, 9, \$C8, 9, \$A8
78B8	09CC024402	DB	9, \$CC, 2, \$44, 2, \$CC, 4, \$44, 5, \$CC, 4, \$44, 3, \$CC, RET
78C7	SUB16	DB	
78C7	02C8034002	DB	2, \$C8, 3, \$40, 2, \$C8, 2, \$40, 3, \$C8, 2, \$40, 5, \$C8, 6, \$A8
78D7	05CC024403	DB	5, \$CC, 2, \$44, 3, \$CC, 2, \$44, 2, \$CC, 3, \$44, 2, \$CC, RET
78E6	SUB17	DB	
78E6	02C8024004	DB	2, \$C8, 2, \$40, 4, \$C8, 4, \$88, 4, \$CC, 2, \$44, 2, \$CC, RET
78E5	SUB19	DB	
78F5	04C00A4009	DB	4, \$C0, 10, \$40, 9, \$C0, 4, \$40, 11, \$C0, 3, \$80, 2, \$C0, 7, \$A0
7905	02C403800B	DB	2, \$C4, 3, \$80, 11, \$C4, 4, \$44, 9, \$C4, 10, \$44, 4, \$C4, RET
7914	SUB20	DB	
7914	03C0044005	DB	3, \$C0, 4, \$40, 5, \$C0, 4, \$40, 2, \$C0, 2, \$40, 9, \$C0, 9, \$A0
7924	09C4024402	DB	9, \$C4, 2, \$44, 2, \$C4, 4, \$44, 5, \$C4, 4, \$44, 3, \$C4, RET
7933	SUB21	DB	
7933	02C4034002	DB	2, \$C4, 3, \$40, 2, \$C0, 2, \$40, 3, \$C0, 2, \$40, 5, \$C0, 4, \$A0
7943	05C4024403	DB	5, \$C4, 2, \$44, 3, \$C4, 2, \$44, 2, \$C4, 3, \$44, 2, \$C4, RET
7952	SUB22	DB	
7952	02C0024004	DB	2, \$C0, 2, \$40, 4, \$C0, 4, \$80, 4, \$C4, 2, \$44, 2, \$C4, RET
7961	SUB9	DB	
7961	10C8	DB	16, \$C8
7963	10C8	DB	16, \$C8
7965	10EA	DB	16, \$EA
7967	1062	DB	16, \$62
7969	1062	DB	16, \$62
796B	1072	DB	16, \$72
796D	1072	DB	16, \$72
796F	1072	DB	16, \$72

DATA TABLES

7971	0872	DB	8,\$72
7973	0862	DB	8,\$62
7975	108084	DB	16,\$90,RET
7978	SUB10		
7978	0864	DB	8,\$64
797A	0875	DB	8,\$75
797C	1075	DB	16,\$75
797E	10FD	DB	16,\$FD
7980	10FD	DB	16,\$FD
7982	10EC	DB	16,\$EC
7984	1064	DB	16,\$64
7986	30EC82	DB	48,\$E4,KILL
7989	SUB11		
7989	10E0	DB	16,\$C0
798B	10C0	DB	16,\$C0
798D	10E2	DB	16,\$E2
798F	1062	DB	16,\$62
7991	1062	DB	16,\$62
7993	1072	DB	16,\$72
7995	1072	DB	16,\$72
7997	1072	DB	16,\$72
7999	0872	DB	8,\$72
799B	0862	DB	8,\$62
79A0	108884	DB	16,\$88,RET
79A0	SUB12		
79A0	0864	DB	8,\$64
79A2	0875	DB	8,\$75
79A4	1075	DB	16,\$75
79A6	10F5	DB	16,\$F5
79A8	10F5	DB	16,\$F5
79AA	10E4	DB	16,\$E4
79AC	1064	DB	16,\$64
79AE	30EC82	DB	48,\$EC,KILL
79B1	SUB1		
79B1	036401C480	DB	3,\$64,1,\$C4,JUMP

; 2\*4:1 LINE STARTING W/GROW

79B1  
79B1 036401C480

SUB1

DB

3,\$64,1,\$C4,JUMP

; 2\*4:1 LINE STARTING W/GROW

DATA TABLES

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 83  
J:5200 A65

79B4 BC79

DW

SUB4

79B5

SUB2

79B8

034401C4

DB

3,\$44,1,\$C4

79B9

SUB4

79B0

034401C484

DB

3,\$44,1,\$C4,RET

; 4:1 LINE

79C1

SUB5

; 2\*4:1 LINE STARTING W/GROW

79C1

036401CC80

DB

3,\$64,1,\$CC,JUMP

79C6

CC79

DW

SUB6

79C8

034401CC

DB

3,\$44,1,\$CC

79C9

SUB8

DB

3,\$44,1,\$CC,RET

; 4:1 LINE

79C0

034401CC84

DB

3,\$44,1,\$CC,RET

; 4:1 LINE

DATA TABLES

1 PICTURE TYPE TO PICTURE CODE CONVERSION TABLES  
2 A NUMBER FROM 0-9 WILL INDEX A GROUP OF 4, AND ANOTHER  
3 NUMBER FROM 0-3 WILL INDEX ONE OF THE 4 WITHIN THE  
4 SUB-GROUP. 0 IS LARGEST WHILE 9 IS SMALLEST.

79T1	E3F70B1F33	DCODEL	DB	LOW_TPOLE,LOW_TSAUC,LOW_TDROID,LOW_TESHT,LOW_TSHOT,LOW_TZORBA,LOW_TSHOT,LOW_TPSHT,LOW_TSTAR
79DA	79797A7A7A	DCODEH	DB	HIGH_TPOLE,HIGH_TSAUC,HIGH_TDROID,HIGH_TESHT,HIGH_TSHOT,HIGH_TZORBA,HIGH_TSHOT,HIGH_TPSHT,HI
GH_TSTAR				
U_79E3	0101F7F7ED	TPOLE	DB	LOW_POL6,LOW_POL6,LOW_POL5,LOW_POL5,LOW_POL4,LOW_POL3,LOW_POL2,LOW_POL1,LOW_DOT,LOW_DOT
U_79ED	9797969696		DB	HIGH_POL6,HIGH_POL6,HIGH_POL5,HIGH_POL5,HIGH_POL4,HIGH_POL3,HIGH_POL2,HIGH_POL1,HIGH_DOT,HIG
H_DOT				
U_79F7	0000000000	TSAUC	DB	LOW_SAU4,LOW_SAU4,LOW_SAU3,LOW_SAU3,LOW_SAU2,LOW_SAU2,LOW_SAU1,LOW_SAU1,LOW_DOT,LOW_DOT
U_7A01	0000000000		DB	HIGH_SAU4,HIGH_SAU4,HIGH_SAU3,HIGH_SAU3,HIGH_SAU2,HIGH_SAU2,HIGH_SAU1,HIGH_SAU1,HIGH_DOT,HIG
H_DOT				
U_7A0B	0000000000	TDROID	DB	LOW_HOP5,LOW_HOP4,LOW_HOP3,LOW_HOP3,LOW_HOP2,LOW_HOP2,LOW_HOP1,LOW_HOP1,LOW_DOT,LOW_DOT
U_7A15	0000000000		DB	HIGH_HOP5,HIGH_HOP4,HIGH_HOP3,HIGH_HOP3,HIGH_HOP2,HIGH_HOP2,HIGH_HOP1,HIGH_HOP1,HIGH_DOT,HIG
H_DOT				
U_7A1F	0000000000	TESHT	DB	LOW_SHOT3,LOW_SHOT2,LOW_SHOT1,LOW_SHOT1,LOW_SHOT1,LOW_SHOT1,LOW_SHOT1,LOW_SHOT1,LOW_SHOT1,LO
W_DOT				
U_7A29	0000000000		DB	HIGH_SHOT3,HIGH_SHOT2,HIGH_SHOT1,HIGH_SHOT1,HIGH_SHOT1,HIGH_SHOT1,HIGH_SHOT1,HIGH_SHOT1,HIGH
SHOT1,HIGH_DOT				
U_7A33	0000000000	TSHOT	DB	LOW_SHOT6,LOW_SHOT6,LOW_SHOT6,LOW_SHOT6,LOW_SHOT5,LOW_SHOT4,LOW_SHOT3,LOW_SHOT2,LOW_SHOT1,LO
W_DOT				
U_7A3D	0000000000		DB	HIGH_SHOT6,HIGH_SHOT6,HIGH_SHOT6,HIGH_SHOT6,HIGH_SHOT5,HIGH_SHOT4,HIGH_SHOT3,HIGH_SHOT2,HIGH
SHOT1,HIGH_DOT				
U_7A47	0000000000	TZORBA	DB	LOW_ZOR5,LOW_ZOR5,LOW_ZOR4,LOW_ZOR4,LOW_ZOR3,LOW_ZOR3,LOW_ZOR2,LOW_ZOR2,LOW_ZOR1,LOW_DOT
U_7A51	0000000000		DB	HIGH_ZOR5,HIGH_ZOR5,HIGH_ZOR4,HIGH_ZOR4,HIGH_ZOR3,HIGH_ZOR3,HIGH_ZOR2,HIGH_ZOR2,HIGH_ZOR1,HI
GH_DOT				
U_7A5B	0000000000	TPSHT	DB	LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LOW_SHOT3,LO
W_SHOT3				
U_7A65	0000000000		DB	HIGH_SHOT3,HIGH_SHOT3,HIGH_SHOT3,HIGH_SHOT3,HIGH_SHOT3,HIGH_SHOT3,HIGH_SHOT3,HIGH_SHOT3,HIGH
SHOT3,HIGH_SHOT3				
U_7A6F	0000000000	TSTAR	DB	LOW_STER,LOW_STER,LOW_STER,LOW_STER,LOW_STER,LOW_STER,LOW_STER,LOW_STER,LOW_STER,LOW_STER
U_7A79	0000000000		DB	HIGH_STER,HIGH_STER,HIGH_STER,HIGH_STER,HIGH_STER,HIGH_STER,HIGH_STER,HIGH_STER,HIGH_STER,HI
GH_STER				
U_7A83	25A9B001E5	MCODEL	DB	LOW_MPOL6,LOW_MSAUC,LOW_MRROID,LOW_MESHT,LOW_MSHOT,LOW_MZORBA,LOW_MSHOT,LOW_MPSHT,LOW_MSTAR
U_7A8C	7A7A7A7A7A	MCODEH	DB	HIGH_MPOL6,HIGH_MSAUC,HIGH_MRROID,HIGH_MESHT,HIGH_MSHOT,HIGH_MZORBA,HIGH_MSHOT,HIGH_MPSHT,HI
GH_MSTAR				
U_7A95	0000000000	MPOLE	DB	LOW_MPOL6,LOW_MPOL6,LOW_MPOL5,LOW_MPOL5,LOW_MPOL4,LOW_MPOL3,LOW_MPOL2,LOW_MPOL1,LOW_MDOT,LOW
MDOT				
U_7A9F	0000000000		DB	HIGH_MPOL6,HIGH_MPOL6,HIGH_MPOL5,HIGH_MPOL5,HIGH_MPOL4,HIGH_MPOL3,HIGH_MPOL2,HIGH_MPOL1,HIGH
MDOT,HIGH_MDOT				
U_7AA9	0000000000	MSAUC	DB	LOW_MSAU4,LOW_MSAU4,LOW_MSAU3,LOW_MSAU3,LOW_MSAU2,LOW_MSAU2,LOW_MSAU1,LOW_MSAU1,LOW_MDOT,LOW
MDOT				
U_7AB3	0000000000		DB	HIGH_MSAU4,HIGH_MSAU4,HIGH_MSAU3,HIGH_MSAU3,HIGH_MSAU2,HIGH_MSAU2,HIGH_MSAU1,HIGH_MSAU1,HIGH
MDOT,HIGH_MDOT				
U_7ABD	0000000000	MRROID	DB	LOW_MHOP5,LOW_MHOP4,LOW_MHOP3,LOW_MHOP3,LOW_MHOP2,LOW_MHOP2,LOW_MHOP1,LOW_MHOP1,LOW_MDOT,LOW
MDOT				
U_7AC7	0000000000		DB	HIGH_MHOP5,HIGH_MHOP4,HIGH_MHOP3,HIGH_MHOP3,HIGH_MHOP2,HIGH_MHOP2,HIGH_MHOP1,HIGH_MHOP1,HIGH

U 7ABD 0000000000 MDROID DB LOW\_MHOP5, LOW\_MHOP4, LOW\_MHOP3, LOW\_MHOP3, LOW\_MHOP2, LOW\_MHOP2, LOW\_MHOP1, LOW\_MHOP1, LOW\_MDOT, LOW\_MDOT

U 7AC7 0000000000 DB HIGH\_MHOP5, HIGH\_MHOP4, HIGH\_MHOP3, HIGH\_MHOP3, HIGH\_MHOP2, HIGH\_MHOP2, HIGH\_MHOP1, HIGH\_MHOP1, HIGH\_MDOT, HIGH\_MDOT

U 7AD1 0000000000 MESHT DB LOW\_MSHOT3, LOW\_MSHOT2, LOW\_MDOT, LOW\_MDOT, LOW\_MDOT, LOW\_MDOT, LOW\_MDOT, LOW\_MDOT, LOW\_MDOT

U 7AD2 0000000000 DB HIGH\_MSHOT3, HIGH\_MSHOT2, HIGH\_MDOT, HIGH\_MDOT, HIGH\_MDOT, HIGH\_MDOT, HIGH\_MDOT, HIGH\_MDOT, HIGH\_MDOT, HIGH\_MDOT

U 7AE5 0000000000 MSHOT DB LOW\_MSHOT6, LOW\_MSHOT6, LOW\_MSHOT6, LOW\_MSHOT6, LOW\_MSHOT5, LOW\_MSHOT4, LOW\_MSHOT3, LOW\_MSHOT2, LOW\_MSHOT1, LOW\_MDOT

U 7AEF 0000000000 DB HIGH\_MSHOT6, HIGH\_MSHOT6, HIGH\_MSHOT6, HIGH\_MSHOT6, HIGH\_MSHOT5, HIGH\_MSHOT4, HIGH\_MSHOT3, HIGH\_MSHOT2, HIGH\_MSHOT1, HIGH\_MDOT

U 7AE9 0000000000 MZORBA DB LOW\_MZOR5, LOW\_MZOR5, LOW\_MZOR4, LOW\_MZOR4, LOW\_MZOR3, LOW\_MZOR3, LOW\_MZOR2, LOW\_MZOR2, LOW\_MZOR1, LOW\_MDOT

U 7B03 0000000000 DB HIGH\_MZOR5, HIGH\_MZOR5, HIGH\_MZOR4, HIGH\_MZOR4, HIGH\_MZOR3, HIGH\_MZOR3, HIGH\_MZOR2, HIGH\_MZOR2, HIGH\_MZOR1, HIGH\_MDOT

U 7B0D 0000000000 MPSHT DB LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3, LOW\_MSHOT3

U 7B17 0000000000 DB HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3, HIGH\_MSHOT3

U 7B21 0000090000 MSTAB DB LOW\_MSTER, LOW\_MSTER

U 7B2B 0000000000 DB HIGH\_MSTER, HIGH\_MSTER

1  
2  
3  
4 : PATTERN TABLES FOR SCROLLING  
5  
6 : EACH DATA BYTE REPRESENTS A ROW THAT CONTAINS A STRIPE.  
7 : 255 ENTRIES FORCE BORDER COLOR USED FOR SHORT ROWS.  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

7B35 0606070707 SCROL0 DB \$06,\$06,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07  
7B44 0707070707 DB \$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06  
7B54 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07  
7B64 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7B74 0706040607 DB \$07,\$06,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
= 0050 SCRELEN EQU \*-SCROL0  
7B85 0606070707 SCROL1 DB \$06,\$06,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$06,\$07,\$07  
7B94 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7BA4 0606070707 DB \$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7BB4 0606070707 DB \$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7BC4 0707070707 DB \$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7BD5 0607060707 SCROL2 DB \$06,\$07,\$06,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07  
7BE4 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7BF4 0707060607 DB \$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7C04 0707070606 DB \$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7C14 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07  
7C25 0607060707 SCROL3 DB \$06,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07  
7C34 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07  
7C44 0707070706 DB \$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7C54 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7C64 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06,\$06  
7C75 0607070607 SCROL4 DB \$06,\$07,\$06,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07  
7C84 0706070707 DB \$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07  
7C94 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07,\$07,\$07  
7CA4 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07  
7CB4 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
7CC5 0607070706 SCROL5 DB \$06,\$07,\$07,\$06,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07  
7CD4 0707070607 DB \$07,\$07,\$06,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$07,\$07,\$07  
7CE4 0707070707 DB \$07,\$02,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$07,\$07,\$07,\$07  
7CF4 0707070707 DB \$07,\$02,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$06,\$06,\$06  
7D04 0707070707 DB \$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07,\$07  
= 7D15 SCREND EQU \*  
7D15 0000000001 D4 DB \$00,\$00,\$00,\$00,\$01,\$01,\$01,\$02,\$02,\$02,\$02,\$03,\$03,\$03  
7D25 0404040405 DB \$04,\$04,\$04,\$04,\$05,\$05,\$05,\$06,\$06,\$06,\$06,\$07,\$07,\$07  
7D35 0808080809 DB \$08,\$08,\$08,\$08,\$09,\$09,\$09,\$09,\$0A,\$0A,\$0A,\$0A,\$0B,\$0B,\$0B  
7D45 0C0C0C0C0C0C DB \$0C,\$0C,\$0C,\$0C,\$0D,\$0D,\$0D,\$0D,\$0E,\$0E,\$0E,\$0E,\$0F,\$0F  
7D55 1010101011 DB \$10,\$10,\$10,\$10,\$11,\$11,\$11,\$12,\$12,\$12,\$12,\$13,\$13  
7D65 1414141415 DB \$14,\$14,\$14,\$14,\$15,\$15,\$15,\$16,\$16,\$16,\$16,\$17,\$17,\$17  
7D75 1818181819 DB \$18,\$18,\$18,\$18,\$19,\$19,\$19,\$19,\$1A,\$1A,\$1A,\$1A,\$1B,\$1B,\$1B  
7D85 1C1C1C1C1C1L DB \$1C,\$1C,\$1C,\$1C,\$1D,\$1D,\$1D,\$1D,\$1E,\$1E,\$1E,\$1F,\$1F  
7D95 2020202021 DB \$20,\$20,\$20,\$20,\$21,\$21,\$21,\$22,\$22,\$22,\$23,\$23,\$23  
7DA5 2424242425 DB \$24,\$24,\$24,\$24,\$25,\$25,\$25,\$26,\$26,\$26,\$27,\$27,\$27  
7DB5 E8E8E8E8E9 DB \$E8,\$E8,\$E8,\$E8,\$E9,\$E9,\$E9,\$EA,\$EA,\$EA,\$EB,\$EB,\$EB  
7DC5 ECECECECED DB \$EC,\$EC,\$EC,\$EC,\$ED,\$ED,\$ED,\$EE,\$EE,\$EE,\$EF,\$EF,\$EF  
7DD5 F0F0F0F0F1 DB \$F0,\$F0,\$F0,\$F0,\$F1,\$F1,\$F1,\$F1,\$F2,\$F2,\$F2,\$F3,\$F3,\$F3  
7DE5 F4F4F4F4F5 DB \$F4,\$F4,\$F4,\$F4,\$F5,\$F5,\$F5,\$F6,\$F6,\$F6,\$F7,\$F7,\$F7  
7DF5 F8F8F8F8F9 DB \$F8,\$F8,\$F8,\$F8,\$F9,\$F9,\$F9,\$F9,\$FA,\$FA,\$FA,\$FB,\$FB,\$FB

7DB5	E8E8E8E8E8E9	DB	\$E8, \$E8, \$E8, \$E8, \$E9, \$E9, \$E9, \$EA, \$EA, \$EA, \$EB, \$EB, \$EB
7DC5	ECECECECED	DB	\$EC, \$EC, \$EC, \$EC, \$ED, \$ED, \$ED, \$EE, \$EE, \$EE, \$EF, \$EF, \$EF
7DD5	F0E0E0F0F0F1	DB	\$EO, \$EO, \$EO, \$EO, \$E1, \$E1, \$E1, \$E2, \$E2, \$E2, \$E3, \$E3, \$E3
7DE5	F4F4F4F4F5	DB	\$F4, \$F4, \$F4, \$F4, \$F5, \$F5, \$F5, \$F5, \$F6, \$F6, \$F6, \$F7, \$F7, \$F7
7DF5	F8F8F8F8F9	DB	\$F8, \$F8, \$F8, \$F8, \$F9, \$F9, \$F9, \$FA, \$FA, \$FA, \$FA, \$FB, \$FB, \$FB

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 86

J:5200 .A65

DATA TABLES

7E05	FCECECECEC	DB	\$EC, \$EC, \$EC, \$EC, \$ED, \$ED, \$ED, \$EE, \$EE, \$EE, \$EE, \$EE
7E15	0002040600	R4	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E25	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E35	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E45	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E55	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E65	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E75	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E85	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7E95	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7EA5	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7EB5	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7EC5	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7ED5	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7EE5	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7EF5	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06
7F05	0002040600	DB	\$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06, \$00, \$02, \$04, \$06

• MUST BE ON 16K BOUNDARY

7QWAZZO	EQU	#\$FF
	ORG	HIGH QWAZZO*\$100

7E15	P1L		
7E15	F018406890	DB	\$F0, \$18, \$40, \$68, \$90, \$B8, \$E0, \$08, \$30, \$58, \$80, \$A8, \$D0, \$F8, \$20, \$48
7F25	7098C0E810	DB	\$70, \$98, \$C0, \$E8, \$10, \$38, \$60, \$B8, \$B0, \$D8, \$00, \$28, \$50, \$78, \$A0, \$C8
7F35	F018406890	DB	\$F0, \$18, \$40, \$68, \$90, \$B8, \$E0, \$08, \$30, \$58, \$80, \$A8, \$D0, \$F8, \$20, \$48
7F45	7098C0E810	DB	\$70, \$98, \$C0, \$E8, \$10, \$38, \$60, \$B8, \$B0, \$D8, \$00, \$28, \$50, \$78, \$A0, \$C8
7F55	F018406890	DB	\$F0, \$18, \$40, \$68, \$90, \$B8, \$E0, \$08, \$30, \$58, \$80, \$A8, \$D0, \$F8, \$20, \$48
7F65	7098C0E810	DB	\$70, \$98, \$C0, \$E8, \$10, \$38, \$60, \$B8, \$B0, \$D8, \$00, \$28, \$50, \$78, \$A0, \$C8
7F74	5078A0C8F0	DB	\$50, \$78, \$A0, \$C8, \$F0, \$18, \$40, \$68, \$90, \$B8, \$E0, \$08, \$30, \$58, \$80, \$A8
7F84	D0F8204870	DB	\$D0, \$F8, \$20, \$48, \$70, \$98, \$C0, \$E8, \$10, \$38, \$60, \$B8, \$B0, \$D8, \$00, \$28
7F94	5078A0C8F0	DB	\$50, \$78, \$A0, \$C8, \$F0, \$18, \$40, \$68, \$90, \$B8, \$E0, \$08, \$30, \$58, \$80, \$A8
7EA4	D0	DB	\$D0

7EA5	Q05Q0A0	SCRTEL	DB	LOW RELSCR, LOW, (RELSCR+SCRLEN), LOW, (RELSCR+(SCRLEN*2))
7EAR	F04090	DB	LOW (RELSCR+(SCRLEN*3)), LOW (RELSCR+(SCRLEN*4)), LOW (RELSCR+(SCRLEN*5))	
7EAR	101010	SCRTRH	DB	HIGH RELSCR, HIGH (RELSCR+SCRLEN), HIGH (RELSCR+(SCRLEN*2))

7FAE	101111	DB	HIGH (RELSCR+(SCRLEN*3)), HIGH (RELSCR+(SCRLEN*4)), HIGH (RELSCR+(SCRLEN*5))
------	--------	----	--

7FB1	P1H		
7FB1	2021212121	DB	\$20, \$21, \$21, \$21, \$21, \$21, \$22, \$22, \$22, \$22, \$22, \$23, \$23

7FC1	2323232324	DB	\$23, \$23, \$23, \$23, \$24, \$24, \$24, \$24, \$24, \$25, \$25, \$25, \$25, \$25, \$25
------	------------	----	--

7FD1	2526242626	DB	\$25, \$26, \$26, \$26, \$26, \$26, \$26, \$27, \$27, \$27, \$27, \$27, \$27, \$27, \$28, \$28
------	------------	----	--

7FE1	2828282829	DB	\$28, \$28, \$28, \$28, \$29, \$29, \$29, \$29, \$29, \$29, \$29, \$2A, \$2A, \$2A, \$2A, \$2A, \$2A
------	------------	----	--

7FF1	2A2B2B2B2B	DB	\$2A, \$2B, \$2B, \$2B, \$2B, \$2B, \$2C, \$2C, \$2C, \$2C, \$2C, \$2C, \$2D, \$2D
------	------------	----	--

8001	2D2B2D2D2E	DB	\$2D, \$2B, \$2D, \$2D, \$2E, \$2E, \$2E, \$2E, \$2E, \$2F, \$2F, \$2F, \$2F, \$30, \$30
------	------------	----	--

8010	3030303030	DB	\$30, \$30, \$30, \$30, \$31, \$31, \$31, \$31, \$31, \$31, \$32, \$32, \$32, \$32, \$32, \$32
------	------------	----	--

8020	3232333333	DB	\$32, \$32, \$33, \$33, \$33, \$33, \$33, \$34, \$34, \$34, \$34, \$34, \$34, \$34, \$35, \$35
------	------------	----	--

8030	3535353535	DB	\$35, \$35, \$35, \$35, \$35, \$36, \$36, \$36, \$36, \$36, \$36, \$37, \$37, \$37, \$37, \$37
------	------------	----	--

8040	37	DB	\$37
------	----	----	------

8041	DLIST =8041	MSG	DLIST =1,*
------	-------------	-----	------------

8041	DLIST	DB	\$70, \$70, \$70
------	-------	----	------------------

8044	462820	DB	\$46, \$28, \$20
------	--------	----	------------------

8047	0646642004	DB	\$04, \$46, \$64, \$20, \$04, \$04, \$84, \$0E, \$8E, \$0E, \$0E, \$8E, \$0E, \$8E, \$0E
------	------------	----	--

8057	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
------	------------	----	--

8067	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
------	------------	----	--

8077	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
------	------------	----	--

8087	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
------	------------	----	--

8097	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
------	------------	----	--

DATA TABLES

80A7	8E0E8E0ECE	DB	\$8E, \$0E, \$8E, \$0E, \$CE, \$00, \$30, \$0E, \$8E
80B0	0E8E0E8E0E	DB	\$0E, \$8E, \$0E, \$3E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
80B8	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
80C8	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
80DB	8E0E8E0E8E	DB	\$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E, \$8E, \$0E
80E9	4180	DW	DLIST
		MSG	'AND ENDS AT ',*

'AND ENDS AT 80EB'

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

&lt;p

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 88  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J: MOUNT .A65

A1	67C8	30/53	30/58	31# 2							
AD1625	73AA	69#49									
ADD0910	73B5	69#54	69#56								
AUD0R	006B	1#16	30/13	30/18	30/23	31/ 3	32/19	32/34			
		32/39	33/19	51/41	51/47	53/26	53/32	53/34			
		53/37	67/24	67/26	67/38	67/40	67/53	67/55			
		68/ 7	68/ 9								
ADDRM	006D	1#17	30/15	30/20	32/31	32/36	51/58	52/ 4			
		67/28	67/30	67/42	67/44	67/57	67/59	68/11			
		68/13									
ADDSCR	7389	18/19	20/40	25/42	63/38	69#30					
ARTS	AE13	56#30									
AUDC1	E801	3#17	14/26	16/30	17/ 4	21/25	23/49	25/ 8			
		37/48									
AUDC2	E803	3#19	21/27	23/50	37/49						
AUDC3	E805	3#21	23/51	25/ 9	37/50						
AUDC4	E807	3#23	23/52	37/51							
AUDCT1	E808	3#14									
AUDF1	E800	3#16	16/17	17/ 7	21/21	21/42	25/ 2	25/30			
AUDF2	E802	3#18	21/45								
AUDF3	E804	3#20	25/27								
AUDF4	E806	3#22									
BDEADC	006A	1#15	17/54	23/ 5	23/20						
BDIN10	6D40	48#36	48/46								
BDIN20	6D43	48#39	48/42								
BDINIT	6D25	37/32	48#23								
BEFLLOC	80EB	36/ 2	87#11	87/50							
BEXPL	74DD	73/ 7	73#22								
BFIRE	7536	73/ 6	73#32								
BFIRWN	7544	73/12	73#34								
BIT	006E	1#18	30/11	32/27							
BLNKIT	634D	15/34	15#37								
BONUS	74C6	73/15	73#17								
BOOML	6659	25#10	25/12	25/32							
BOOZ	665B	25#11	25/14								
BORING	65F4	23#53	23/56								
BOUNCE	94BB	87#56									
BR1	72BA	65/61	68#30								
BSERAS	7232	23/ 8	43/16	67#23							
BSEFCNT	0081	1#36	45/11	45/13	45/17	45/42					
BSEI10	6CA9	45/10	45#15								
BSEI20	6CAE	45/12	45#22								
BSEI30	6CDE	45/26	45#45								
BSFIRE	6C9C	4/22	45# 7	68/19							
BSOUND	639B	4/38	17# 2								
BSPAIN	726E	44/58	67#52								
BSW100	6C96	44/11	44/32	44/40	44/42	44/50	44#54				
BSW020	6C35	43#36	43/50	43/61							
BSW030	6C3A	43/19	43#42								
BSW040	6C4A	43/21	43#54								
BSW050	6C57	43/25	43/38	43/49	44# 5						
BSW060	6CAF	44/ 8	44#24								
BSW070	6C78	44/20	44#30								
BSW080	6C7D	44/29	44#36								
BSW090	6C8A	44/19	44#46								
BSWORK	6C23	4/25	43#15								
BUCK	969D	68/24	68/25	87#56							
BUCKL	96A7	68/24	68/25	87#56							
BUCKR	96B1	68/24	68/25	87#56							
BUCKX	009A	1#57	22/14	22/54	37/41	43/37	43/45	43/57			

1	BUCKY 009B	45/27	63/23	67/31	67/45	67/60	68/14	
2		1#58	22/16	22/56	37/89	44/15	44/25	44/31
3		45/31	67/47	68/18	72/32			
4	BUKSH 72E1	67/39	68/13	68/25				
5	BUKSL 72AE	67/37	68/6	68/24				
6	RYERYE 635D	15/36	15/38	15/42	15#44			
7	RYTE 0070	1#19	30/9	30/29	30/31	30/52	30/54	30/60
8		31/20	32/25	32/45	32/47	33/8	33/12	33/16
9		33/40						
10	CALL 0083	73#58	74/50	74/52	74/54	75/7	75/10	75/13
11		75/17	75/21	75/23	75/26	75/28	75/32	75/34
12		75/44	75/47	75/50	75/54	75/56	75/60	76/3
13		76/5	76/9	76/11	76/21	76/23	76/28	76/30
14		76/35	76/37	76/42	76/44	77/8	77/10	77/12
15		77/14	77/18	77/20	77/22	77/24	77/28	80/28
16		80/35	80/42	80/49	80/59	81/6	81/13	
17	CHACTL D401	3#9	71/19					
18	CHART 0904	2#46	35/41	71/18				
19	CHBAS 0903	2#45	23/42	35/51	45/43	71/20		
20	CHBASE D409	3#10	71/21					
21	CHKH10 7367	68#60	69/7					
22	CHKH20 737A	49/4	49#8					
23	CHKH30 737B	69#10	69/14					
24	CHKH40 7381	69/3	69#15					
25	CHKH50 7365	35/48	68#58					
26	CHKITA 722B	67/10	67#14					
27	CHKVAL 6519	21#52	21/54					
28	CINIT 68E5	4/3	4/16	23/57	35#7			
29	CKCR10 66CC	26/36	26#42					
30	CKCR20 66D0	26/41	26#45					
31	CKCR30 66F3	26/49	26#55					
32	CKCR40 66E7	26/54	26#58					
33	CKCR50 66EA	26/40	26#44	26/53	26/57	26#61		
34	CKCR5H 66AD	20/21	20/29	20/44	26#26			
35	CLERHI 6015	4#7	4/11					
36	CLEV1 60E4	49/55	49/56	49#57				
37	CLEV2 60E8	49/55	49/56	49#58				
38	CLEV3 60EA	49/55	49/56	49#59				
39	CLEV4 60EC	49/55	49/56	49#60				
40	CLEV5 60EE	49/55	49/56	49#61				
41	CLOOP 6956	36#2	36/5					
42	COL110 6487	20#37						
43	COL114 64C1	20/55	21#5					
44	COL115 64CC	21/4	21#14					
45	COL120 6530	19/36	22#13					
46	COL130 6542	22#23	22/32					
47	COL140 6547	22#26	22/37	22/39	22/41	22/43	22/45	
48	COL145 6550	22#33						
49	COL150 6551	22/25	22#35					
50	COL160 6595	23#9	23/21					
51	COL170 65A2	23#15	23/17	23/19				
52	COL180 65C1	23/27	23#32					
53	COL185 65C4	23/23	23#34					
54	COL190 6600	20/23	24#8					
55	COL195 661B	24/20	24#24					
56	COL200 661C	24/14	24#31					
57	COL210 662B	24#37						
	COL220 6643	24/26	24#52					
	COL260 6693	25#37						
	COL280 6695	25#39	25/44					

COL220 6643 24/24 24#52  
n COL260 6693 25#37  
COL280 6695 25#39 25/44

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 90  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J: MOUNT .A65

COLDEG 0907	2#49	3/57	3/60	4/13	4/15
COLDET 6011	3/59	4/2	4# 4		
COLI10 6428	19#25	19/35			
COLI20 642E	19#29	20/19			
n COLI25 643B	19#37				
COLI30 643C	19/28	19#42			
COLI40 645D	20# 2	20/16			
COLI50 646D	20#10	20/22	20/30		
COLI60 6475	20#17	21/ 9	24/48		
COLI65 647A	20/ 9	20#20			
COLI70 6482	20/ 5	20/ 7	20#28		
COLISN 6426	4/29	19#23			
COLOR0 0911	2#52	21/30	23/54	25/22	48/25 66/50 70/55
COLOR1 0912	2#53	21/32	23/45	25/16	48/27 66/52 70/57
COLOR2 0913	2#54	21/34	23/46	25/18	48/29 66/56 70/59
COLOR3 0914	2#55	15/44	21/36	23/47	25/20 48/30 66/54
	67/ 7	70/61			
COLOR4 0915	2#56	21/38	23/48	25/25	48/31 66/ 9 71/ 3
COLORSH 60E1	49/26	49#54			
COLRSL 610D	49/24	49#55			
CONFGT 7029	61/ 8	61/10	61#14		
CONSOL 001F	3#15	71/ 6			
CONVOK 607D	4/47	4/48	4#52		
CONVSP 606A	4/19	4#41	23/ 3	37/45	44/57
CRASH 750C	73/ 5	73#26			
CREA10 60E6	45/25	46#16			
CREA20 60F3	46/15	46#23			
CREA30 6CE6	46/17	46#26			
CREA40 6CEF	46#32	46/36			
CREA5 6CE2	46#13	46/22			
CREATE 6CEO	6/18	7/27	8/15	9/36	10/52 12/16 22/52
	28/11	28/15	42/39	46#11	
DSCORE 090D	2#51	47/ 9	18/24	18/35	36/24 36/25 36/26
	36/27	65/10	65/19	66/31	66/40 68/61 69/11
	69/33	69/34	69/36	69/37	69/39 69/40 69/42
	69/43				
CVLOOP 6071	4#44	4/50			
DCODEH 79DA	51/36	53/21	84# 8		
DCODEL 79D1	51/34	53/19	84# 7		
DEADBK 656Z	22#49	22/53	72/37		
DECII 64BD	20/52	21# 3			
DLILOC 00B1	2#30	35/32	35/34	36/ 3	87/12
n DLIRITN 00B1	87#14				
BLIST 8041	35/54	35/56	86#52	87/ 6	
n BLISTH D403	3#12				
n BLISIL D402	3#11				
DMACTL D400	3#13	71/12			
U.DOT 79E3	84/10	84/10	84/11	84/11	84/13 84/13 84/14
	84/14	84/16	84/16	84/17	84/17 84/19 84/20
	84/22	84/23	84/25	84/26	
DRAW 67EC	31#26	31#34			
DRIVER 602B	4#17	4/39	38/ 8		
ERASE 6818	32#17	55/41	67/35	67/49	
EXITLC 67B9	30/32	30/33	30#54		
n EXITRB 722A	67#13				
FIRDLY 00AF	2#28	36/11	45/41		
FIRFDG 00B0	2#29	19/48	19/54	36/14	
FRGTTT 6897	33/26	33#28	33/29		
FUEL 62C9	4/32	14# 7			
n FUEL03 62C9	14#11				

FUEL05	6201	14#16	14/34					
FUEL07	6202	14/15	14#18					
FUEL10	6801	14/38	14#43	39#4				
FUELST	7560	73/41	73#40					
FUJISOK	630C	14/55	14#57					
FULAMT	009C	1#59	14/12	14/13	14/14	14/27	14/30	14/31
		14/33	14/35	14/37	14/40	14/41	14/42	14/50
		15/ 7	15/26	23/24	23/25	23/26	37/19	37/21
		37/22	71/47	71/48	71/49			
FULP	6312	14#61	15/ 3					
FLIXI	6335	15/15	15#26					
FLIXX	6328	15/ 9	15#14	15/19				
GETNXT	73B4	61/30	61/37	61/44	70# 2			
GL1	67A5	30/41	30#43					
GL2	67B3	30/49	30#51					
GL3	6802	31/36	31#38					
GL4	6807	31/42	31#44					
GL5	6812	31/48	31#50					
GLOPO	6647	25# 2	25/ 4					
GLZ10	600CF	49/46	49#48					
GLZ9	60029	49/43	49#45					
GRACTL	C01D	3#24	35/31					
HERE	6204	31#14	31/52					
HFIRE	7552	73/ 8	73#36					
HFRCNT	0090	1#45	9/ 9	9/11	9/55			
HFRDLY	00A9	2#15	9/54					
HIGHT	0071	1#20	31/ 4	31/51	33/20	34/10		
HISCOR	0909	2#50	4/ 8	66/11	66/20	69/ 2	69/12	
U HOP1	7AQB	84/16	84/16	84/17	84/17			
U HOP2	7AQB	84/16	84/16	84/17	84/17			
U HOP3	7AQB	84/16	84/16	84/17	84/17			
U HOP4	7AQB	84/16	84/17					
U HOP5	7AQB	84/16	84/17					
HOPCNT	0084	1#39	12/ 9	12/11	12/33			
HOPDLY	00A3	2# 9	12/32					
HOPF10	61A4	9/ 8	9#12	9/37				
HOPF20	61A5	9/10	9#14					
HOPF30	61A7	9#16	9/26					
HOPF35	61AE	9#20	9/31	9/33				
HOPF40	61BA	9/19	9#28					
HOPF50	61F4	9/27	9#53					
HOPFIR	619A	4/33	9# 6					
HOPP10	628A	12/ 8	12#12	12/17				
HOPP20	628B	12/10	12#14					
n HOPP30	6292	12#18						
n HOPP40	629D	12#24						
n HOPP50	62A8	12#29						
HOPPER	627E	4/28	12# 5					
HOPSPD	00A4	2#10	12/30					
HSCOUT	71AE	66#11	66/28					
n INC01	749E	72#44						
INC010	74A4	72#48	72#50					
INC210	74AD	72/55	72#57					
n INC23	74A7	72#53						
INDR10	73C1	70/10	70#12					
INDRPT	73BB	60/57	62/10	62/11	62/18	70/ 3	70# 8	
n INIT	6904	35#26						
INSD10	7388	69/23	69#25					
INSDPT	7382	16/12	16/24	69#21				
INVDAT	8125	87#56	87/56	87/56	87/56	87/56	87/56	

INSDPT 7382 16/12 16/24 69#21  
INVDAT 8125 87#56 87/56 87/56 87/56 87/56 87/56 87/56

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 92  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J: MOUNT .A65

1	87/56	87/56	87/56	87/56	87/56	87/56	87/56	
2	87/56	87/56	87/56	87/56	87/56	87/56	87/56	
3	87/56	87/56	87/56	87/56	87/56	87/56	87/56	
4	87/56	87/56	87/56	87/56	87/56	87/56	87/56	
5	87/56	87/56	87/56	87/56	87/56	87/56	87/56	
6	JAMIT 6208	13/ 9	13#18					
7	JOY100 749B	72/34	72#36					
8	JOYS10 7443	71/50	71#52					
9	JOYS20 7490	71/51	72#29					
10	JOYSTK 7434	4/18	71#44					
11	JS1 7453	71/55	71#60					
12	JS2 7460	72/ 7	72#12					
13	JSBATA 0060	1#12	40/13	43/18	44/ 6	44/ 9	45/ 8	71/46
14		71/56	71/58	72/ 2	72/ 4	72/ 8	72/10	72/14
15		72/16	72/20	72/22	72/25	72/27	72/31	
16	JSJAM 7477	72/11	72/13	72#17				
17	JSJAM1 7484	72/19	72#23					
18	JSJAM2 748E	72/24	72#28					
19	JUMP 0080	73#55	77/30	77/35	77/39	79/49	79/52	79/55
20		79/58	79/61	80/ 4	80/ 7	80/10	80/13	80/16
21		80/19	80/22	80/30	80/37	80/44	80/51	80/56
22		80/61	81/ 8	81/15	82/60	83/ 8		
23	KBCODE E802	3#36	66/57	67/ 8				
24	KBFLG 007A	1#26						
25	KILL 0082	73#57	74/56	75/37	76/14	77/ 3	79/22	79/44
26		82/19	82/57					
27	LASTBS 0080	1#35	43/24	43/44	43/56	67/36	68/ 5	
28	U LEVELD 6997	36/44						
29	LOADPC 0081	73#56	76/48	80/28	80/35	80/42	80/49	80/56
30		80/59	81/ 6	81/13				
31	LOCEND 00EB	87#43	87/50					
32	LOCSTR 00B1	87#18	87/50					
33	LVLDAT 009E	2# 4	2/19	36/60				
34	LVLFLG 007B	1#27	36/20	55/28	55/31	63/44		
35	LVLQVR 64CC	21#13	55/32					
36	MASK 0072	1#21	30/16	30/21	30/47	30/48	30/50	31/27
37		31/34	31/35	31/37	32/32	32/37	33/ 3	33/ 4
38		33/ 6	33/47	33/53	33/54	33/56		
39	MAXLVL 0014	36/34	39#58					
40	U MBUCK 72B4	68/27	68/28					
41	U MBUCKL 72B4	68/27	68/28					
42	U MBUCKR 72B4	68/27	68/28					
43	MBUKSH 72B7	67/43	68/12	68#28				
44	MBUKSL 72B4	67/41	68/10	68#27				
45	MCODEH 7A8C	51/53	84#36					
46	MCODEL 7A83	51/51	84#35					
47	U MDOT 7A95	84/38	84/38	84/39	84/39	84/41	84/41	84/42
48		84/42	84/44	84/44	84/45	84/45	84/47	84/47
49		84/47	84/47	84/47	84/47	84/47	84/47	84/48
50		84/48	84/48	84/48	84/48	84/48	84/48	84/48
51		84/50	84/51	84/53	84/54			
52	MROID 7ABD	84/35	84/36	84#44				
53	MEANEG 007C	1#28	36/23	36/39	36/42	37/ 9		
54	MESHT 7A01	84/35	84/36	84#47				
55	MFRCNT 0092	1#47	8/ 3	8/ 5	8/10			
56	MFRDLY 00AB	2#17	8/ 9					
57	U MHOP1 7ABD	84/44	84/44	84/45	84/45			
	U MHOP2 7ABD	84/44	84/44	84/45	84/45			
	U MHOP3 7ABD	84/44	84/44	84/45	84/45			



U MSHOT4 7AE5 84/50 84/51  
U MSHOT5 7AE5 84/50 84/51

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 94  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

11 MSHOT6 7AE5 84/50 84/50 84/50 84/50 84/51 84/51 84/51

12 MSTAR 7B21 84/51 84/51 84/51 84/51 84/51 84/51 84/51

13 MATER 7B21 84/59 84/59 84/59 84/59 84/59 84/59 84/59

14 84/59 84/59 84/59 84/60 84/60 84/60 84/60

15 84/60 84/60 84/60 84/60 84/60 84/60 84/60

16 MTN130 6B00 41/ 2 41/27 41#40

17 MTNCNT 0098 1#55 37/31 40/60 40/61 41/ 4

18 MTNSPD 0096 1#53 22/61 37/47 40/18 40/33 40/40 40/46

19 40/52 41/ 5 57/58

20 MTNW05 6B44 40/ 9 40#11

21 MTNW10 6B5C 40/19 40#25

22 MTNW30 6B66 40/14 40#32

23 MTNW40 6B72 40/16 40#39

24 MTNW50 6B7B 40/22 40#24 40/28 40/30 40/36 40/38 40/43

25 40#45

26 MTNW60 6B7D 40#50

27 MTNW70 6B87 40/53 40#57

28 MTNW80 6B96 41/ 6 41#14

29 MTNW90 6B92 41/10 41/13 41/18 41#24 48/60

30 MTNW90 6B9C 41#30 41/39

31 MTNW9K 6B3D 41/20 40# 6

32 MTOK 6B07 41/35 41#37

33 MTRBAK 0087 74# 2 78/19 78/60

34 MTRHTB 66A4 8/23 24/15 25#50

35 MTRSPD 00A5 2#11 7/44

36 MTSTRT 0097 1#54 37/29 41/ 7 41/ 8 41/12 41/15 41/16

37 41/20 41/29

38 MULT 73C2 14/24 36/47 56/46 70#30

39 MULT10 73D0 70#32 70/46

40 MULT20 73D4 70/40 70#42

41 MULT30 78DD 70/32 70#49

42 MVXY 6FD4 57/27 57/39 59# 4

43 MVXY10 6EE0 59#11 59/17 59/25 59/33

44 MVXY20 6EE2 59/ 6 59#13

45 MVXY30 6EE3 59/ 5 59/ 8 59#15

46 MVXY40 6EE8 59/ 9 59#26 59/34

47 MVXY50 6FFB 59/20 59#29

48 U MZOR1 7AF9 84/53 84/54

49 U MZOR2 7AE9 84/53 84/53 84/54 84/54

50 U MZOR3 7AF9 84/53 84/53 84/54 84/54

51 U MZOR4 7AF9 84/53 84/53 84/54 84/54

52 U MZOR5 7AE9 84/53 84/53 84/54 84/54

53 MZORBA 7AF9 84/35 84/36 84#53

54 NEWL05 6997 36/35 36#43

55 NEWL10 69B4 36#58 37/ 3

56 NEWL20 69D5 37/17 37#23

57 NEWLVI 69B5 22/ 5 23/31 25/46 36#32

58 NEWORG 8125 87#50 87/52 87/54

59 NEWP10 6A11 37#54 37/58

60 NEWPLY 69D5 23/33 37#27

61 NMIEN D40F 3#25 23/36 35/11 35/53 42/ 3 49/13 49/17

62 45/45

63 NOGAIN 6E02 55/29 55#33

64 NXT100 70C3 62/52 62#54 63/ 4 63/ 7

65 NXT110 70C6 62/45 62#56

66 NXT120 70C9 62/48 62#59

67 NXT130 70D3 63/ 3 63# 5

68 NXT140 70D8 61/21 63#15

69 NXT150 70E6 63/17 63#22



PATH19 77CF 74/26 79#48 79/50  
PATH2 7661 74/ 9 75#39

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 96  
BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J: MOUNT .A65

1 PATH20 77D4	74/27	79#51	79/53
2 PATH21 77DB	74/28	79#54	79/56
3 PATH22 77E2	74/29	79#57	79/58
4 PATH23 77E7	74/30	80# 6	80/ 2
5 PATH24 77EE	74/31	80# 3	80/ 5
6 PATH25 77E5	74/32	80# 4	80/ 8
7 PATH26 77EA	74/33	80# 9	80/11
8 PATH27 7801	74/34	80#12	80/14
9 PATH28 7808	74/35	80#15	80/17
10 PATH29 780D	74/36	80#18	80/20
11 PATH30 7814	74/37	80#21	80/23
12 PATH31 781B	74/38	80#27	80/31
13 PATH32 782A	74/39	80#34	80/38
14 PATH33 7839	74/40	80#41	80/45
15 PATH34 7848	74/41	80#48	80/52
16 PATH35 7857	74/42	80#55	
17 PATH36 785C	74/43	80#58	81/ 2
18 PATH37 786B	74/44	81# 5	81/ 9
19 PATH38 787A	74/45	81#12	81/16
20 PATH4 76A1	74/11	76#26	
21 PATH5 76A9	74/12	76#33	
22 PATH6 76B1	74/13	76#40	
23 PATH7 76B9	74/14	76#47	
24 PATH8 76D3	74/15	77# 2	
25 PATH9 76C1	74/16	76#52	
26 U PAUSE 6061	4/37		
27 PBCTL D303	3#27	35/44	
28 PCSCL 6E50	12/51	26/31	53#14
29 PCXY05 6DE7	51#25	51/31	
30 PCXY10 6DEA	51/24	51#28	
31 PCXYAC 6DFO	51#21	55/39	55/47
32 PDL0 0916	2#57	71/14	71/53
33 PDL1 0917	2#58	71/16	72/ 5
34 PFIRE 755E	73/ 9	73#38	
35 PFRCNT 0021	1#46	10/ 8	10/10
36 PFDLY 00AA	2#16	10/27	
37 PLYLVL 00AE	2#24	21/15	23/28
38	36/38	36/45	37/15
39 PLYTBL 6A25	36/50	36/54	49/20
40 PMBASE D407	3#28		
41 POL1 96CE	84/10	84/11	87#56
42 POL1110 426F	10/37	11#11	
43 POL2 96D9	84/10	84/11	87#56
44 POL3 96E3	84/10	84/11	87#56
45 POL4 96FD	84/10	84/11	87#56
46 POL5 9AFZ	84/10	84/10	84/11
47 POL6 9701	84/10	84/10	84/11
48 POLCNT 0082	1#37	28/ 8	28/23
49 POLDLY 00A0	2# 6	28/59	
50 POLE10 670E	28/ 7	28/12	28#20
51 POLE20 670E	28/ 9	28#22	
52 POLE30 6714	28/16	28#28	
53 POLE35 6716	28#30		
54 POLE40 6716	28#31	28/34	28/40
55 POLES 66ED	4/26	28# 4	
56 POLF10 6205	10/ 7	10#11	
57 POLE20 6206	10/ 9	10#13	
58 POLF30 6208	10#15	10/25	
59 POLF40 620F	10#19	10/34	10/40
60	11/10	11/14	



57  
 SCOR05 6422 18/34 18/34  
 SCORE 63DB 4/30 17#53 23/12  
 SCOREC 0061 1#13 17/56 18/ 8 18/ 9

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 98  
 BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

SCRATC 008E	1#44	15/31	15/33					
n SCRONT 0095	1#52							
SCREEN 0074	1#72	31/17	31/21	31/23	31/26	31/29	33/37	
	33/41	33/43	33/46	33/48				
SCREND 7D15	49/49	49/52	85#44					
SCRLLAD 0088	87#19	87/33	87/35					
SCRLEN 0050	85#18	86/35	86/35	86/36	86/36	86/36	86/37	
	86/37	86/38	86/38	86/38				
SCRLLFR 00CE	87#31	87/40						
SCRLLT 00B6	49/11	87#17	87/42					
SCROL0 7B35	49/28	49/30	85#13	85/18				
n SCROL1 7B85	85#19							
n SCROL2 7B05	85#24							
n SCROL3 7C25	85#29							
n SCROL4 7C75	85#34							
n SCROL5 7CC5	85#39							
n SCRSET 0094	1#51							
SCRSPD 0097	1#54	4/44	14/19	17/ 5	18/ 4	23/ 2	37/44	
	44/37	44/39	44/41	44/47	44/48	44/52	56/45	
SCRTH 7F4B	86#87	87/34						
SCRTHL 7FA5	86#35	87/32						
SCRU10 7143	45#10	65/27						
SCRUPD 713F	18/42	65# 7						
SDLOP 6D8F	49#14	49/16						
SDLSTH 0901	2#43	23/38	35/57	65/41	71/ 9			
SDLSTL 0900	2#42	23/40	35/55	65/39	71/ 7			
SDMCTL 0902	2#44	65/47	71/11					
SNDSTAT 0089	1#41	16/ 9	16/29	17/19	17/34	37/52		
SETCOL 6D99	37/33	49#20						
SETDOL 6D82	37/37	49# 9						
SETPTH 6D0F	6/40	7/39	8/18	9/39	10/55	12/23	20/44	
	20/49	22/59	24/42	28/55	28/58	42/33	45/40	
47# 7	64/13							
SETSPD 0085	73#60	76/49	77/53					
SHADOW 96C5	67/23	67/25	67/52	67/54	87#56			
SHAPE 007A	1#23	30/14	30/19	30/39	30/40	30/42	31/28	
	31/46	31/47	31/49	32/30	32/35	32/55	32/56	
	32/58	34/ 5	34/ 6	34/ 8				
SHIPS 62B2	13# 5	18/40	37/42					
SHOK 62RE	13/11	13#13						
SHOPD 62C2	13#15	13/17						
U SHOT1 7A1E	84/19	84/19	84/19	84/19	84/19	84/19	84/19	
	84/20	84/20	84/20	84/20	84/20	84/20	84/20	
	84/22	84/23						
U SHOT2 7A1F	84/19	84/20	84/23	84/23				
U SHOT3 7A1F	84/19	84/20	84/23	84/23	84/28	84/28	84/28	
	84/28	84/28	84/28	84/28	84/28	84/28	84/28	
	84/29	84/29	84/29	84/29	84/29	84/29	84/29	
	84/29	84/29	84/29	84/29	84/29	84/29	84/29	
U SHOT4 7A33	84/22	84/23						
U SHOT5 7A33	84/22	84/23						
U SHOT6 7A33	84/22	84/22	84/22	84/22	84/23	84/23	84/23	
	84/23							
SHPLFT 00AD	2#23	13/ 6	18/31	18/33	23/22	36/12		
SKCTL E80E	3#33	35/47						
SKIPMT 6D70	48/59	49# 2						
SKPDLI 69EC	37/36	37#38						
SKSTAT E80E	3#35	72/17						
U SLIST 65C9	23/37	23/39						
SNDAGE 0085	1#40	16/18	16/21	17/36				



SVA 00E7 87/14 87#44  
 SVX 00E9 87/15 87#46  
 SYSTAT 00AC 2#18 6/ 7 7/ 6 7/29 9/ 7 10/ 5 10/38

SORCIM 6502 Assembler ver 3.5F 06/26/84 11:18 Page 100  
 BUCK RODGERS GAME FOR ATARI 400/600XL/800/800XL/1200XL/1400X J:MOUNT .A65

1	11/12	12/ 6	20/50	20/58	20/61	25/ 6	25/26
2	25/31	28/ 5	36/21	37/34	41/25	42/ 7	43/57
3	66/58	67/ 9					
4	TDRDIB 7A0B	84/ 7	84/ 8	84#16			
5	TESHT 7A1F	84/ 7	84/ 8	84#19			
6	TGRN 635B	15/40	15#43				
7	TH 67E7	31/15	31#32				
8	TIDAT 6D7C	48/53	49# 7				
9	TIMER 0078	1#24	23/44	23/53	66/61	67/ 6	71/22
10	TITLE 7163	35/49	65#35				
11	TLIST 7346	65/38	65/40	68#49	68/53		
12	TPOLE 79E3	84/ 7	84/ 8	84#10			
13	TPSHT 7A5B	84/ 7	84/ 8	84#28			
14	TSAUC 79F7	84/ 7	84/ 8	84#13			
15	TSHOT 7A33	84/ 7	84/ 7	84/ 8	84/ 8	84#22	
16	ISTAB 7A6E	84/ 7	84/ 8	84#31			
17	TYEL 6352	15/28	15#39				
18	TZORBA 7A47	84/ 7	84/ 8	84#25			
19	UDSCOL 6D9E	21/55	49#24				
20	UFCONT 00A6	2#12	5/ 5	20/53	20/54	20/57	21/ 8
21	UFODSP 608B	5# 4	21/ 6	38/ 5	43/50		
22	UEGLX 60A1	5#15	5/12	5/21			
23	UEOOK 609B	5#12	5/25				
24	UFOXT 60B0	5/ 6	5/17	5#22			
25	UEPT 60B1	5/ 8	5#23				
26	VRIRTN 73DF	35/36	35/38	70#53			
27	VCOUNT D40B	3#34	21/40	25/11	33/28	49/14	66/59
28	VDISLT 0206	3#29	35/33	35/35			
29	VVBLKI 0202	3#30	35/37	35/39			
30	WAITVB 6504	21#40	21/41	21/47			
31	WARP 64CE	21#19					
32	WARP1 64E4	21#29	22/ 4				
33	WARP5 64D2	21#21	21/23				
34	WIDTH 007F	1#31	30/27	30/35	30/43	31/33	31/38
35		32/43	32/51	32/59	33/52	33/57	34/ 3
36	WIDTHC 007E	1#30	30/34	30/37	30/45	30/55	30/61
37		32/50	32/53	32/61	33/11	33/17	33/44
38	WSYNC D40A	3# 8	67/ 4	87/20			
39	XLOOP 6DB8	49#36	49/50	49/53			
40	XSHIFT 6ECB	57/56	58#32				
41	YAXIS 745D	71/59	71/61	72# 5			
42	YCOOD 0079	1#25	30/ 7	30/26	32/23	32/42	33/21
43	ZA1 6886	33/ 9	33/14	33#18			
44	ZAPIT 68ED	35#13	35/22				
45	ZDRAW 68BB	33#46	33/50				
46	ZEXITL 6877	32/48	32/49	33#10			
47	ZGL1 6863	32/57	32#59				
48	ZGL2 6871	33/ 5	33# 7				
49	ZGL3 68CE	33/55	33#57				
50	ZGL4 68D4	33/61	34# 8				
51	ZGL5 68DE	34/ 7	34# 9				
52	ZHERE 68A3	33#34	34/11				
53	ZOD 6327	15#13					
54	ZODS 631B	15# 7					
55	ZOKP1 68B8	33/42	33#44				
56	U ZOR1 7A47	84/25	84/26				
57	U ZOR2 7A47	84/25	84/25	84/26	84/26		
58	U ZOR3 7A47	84/25	84/25	84/26	84/26		
59	U ZOR4 7A47	84/25	84/25	84/26	84/26		
60	U ZOR5 7A47	84/25	84/25	84/26	84/26		

ZORBD	7577	25/29	73/13	73#43
ZRCLIP	6878	32/46	33#11	
ZTH	6664	33/35	33#51	
ZXCBE	7185	65#50	65/59	
ZXCLP	718A	65#53	65/55	